# Robust RVM Regression Using Sparse Outlier Model

Kaushik Mitra, Ashok Veeraraghavan* and Rama Chellappa
Department of Electrical and Computer Engineering and
Center for Automation Research, UMIACS
University of Maryland, College Park, MD
*Mitshubishi Electric Research Laboratories, Cambridge, MA
{kmitra, vashok, rama}@umiacs.umd.edu

## Abstract

*Kernel regression techniques such as Relevance Vector Machine (RVM) regression, Support Vector Regression and Gaussian processes are widely used for solving many computer vision problems such as age, head pose, 3D human pose and lighting estimation. However, the presence of outliers in the training dataset makes the estimates from these regression techniques unreliable. In this paper, we propose robust versions of the RVM regression that can handle outliers in the training dataset. We decompose the noise term in the RVM formulation into a (sparse) outlier noise term and a Gaussian noise term. We then estimate the outlier noise along with the model parameters. We present two approaches for solving this estimation problem:* 1) *a Bayesian approach, which essentially follows the RVM framework and* 2) *an optimization approach based on Basis Pursuit Denoising. In the Bayesian approach, the robust RVM problem essentially becomes a bigger RVM problem with the advantage that it can be solved efficiently by a fast algorithm. Empirical evaluations, and real experiments on image denoising and age estimation demonstrate the better performance of the robust RVM algorithms over that of the RVM regression.*

## 1. Introduction

Kernel regression techniques such as Support Vector Regression (SVR) [21], Relevance Vector Machine (RVM) regression [17] and Gaussian processes [13] are widely used for solving many vision problems. Some examples are age estimation from facial images [10, 9, 7, 8], head pose estimation [11], 3D human pose estimation [2] and lighting estimation [14]. Recently, kernel regression has also been used for solving image processing problems such as image denoising and image reconstruction with a great deal of success [15, 16]. However, many of these kernel regression methods, especially the RVM, are not robust to outliers in

the training dataset and hence, will produce unreliable estimates in the presence of outliers.

In this paper, we present two robust versions of the RVM regression that can handle outliers in the training dataset. We decompose the noise term in the RVM formulation into an outlier noise term, which we assume to be sparse, and a Gaussian noise term. The assumption of outliers being sparse is justified as we generally expect the majority of the data to be inliers. During inference, we estimate the outlier noise along with the model parameters.

We present two approaches for solving this estimation problem: 1) a Bayesian approach and 2) an optimization approach. In the Bayesian approach, we assume a joint sparse prior for the model parameters and the outliers, and then solve the Bayesian inference problem. The mean of the posterior distribution of the model parameters is used for prediction. The joint sparse assumption for the model parameters and the outliers, effectively, makes the robust RVM problem a bigger RVM problem with the advantage that we can use a fast algorithm, developed for the RVM [18], to solve this problem. In the optimization approach, we attempt to minimize the $L_0$ norm of the model parameters and the outliers, subject to a certain amount of observation error (which depends on the inlier noise variance). However, this minimization is a combinatorial problem; hence, it cannot be solved directly, so we solve a convex relaxation of the problem, known as Basis Pursuit Denoising [4]. We then empirically evaluate the robust algorithms by varying three important intrinsic parameters of the robust regression problem: the outlier fraction, the inlier noise variance and the number of data points in the training dataset. We further demonstrate the effectiveness of the robust approache in solving the image denoising and age estimation problems.

**Prior Work** Robust versions of the RVM regression have been proposed in [6], [19] and [22]. In [6], the noise term is modeled as a mixture of Gaussian (for the inlier noise), and uniform or Gaussian with large variance for the

outlier noise. But the mixture density model makes inference difficult; a variational method is used for solving this problem making it computationally much more expensive than the RVM. In [19], a Student's t-distribution is assumed for the noise, and the parameters of the distribution are estimated along with the model parameters. Though, this is a very elegant approach; a variational method is used for inference, which similar to [6], makes it computationally expensive. In [22], a trimmed likelihood function is minimized over a 'trimmed' subset that does not include the outliers. The robust trimmed subset and the model parameters are found by an iterative re-weighting strategy, which at each iteration solves the RVM regression problem over the current trimmed subset. However, the method needs an initial robust estimate of the trimmed subset, which determines the accuracy of the final solution. It also needs many iterations, where in each iteration a RVM regression problem is solved, and this makes it slow.

## 2. Robust RVM Regression

For both the Bayesian approach and the optimization approach, we replace the Gaussian noise assumption in the RVM formulation by an implicit heavy-tailed distribution. This is achieved by decomposing the noise term into a sparse outlier noise term and a Gaussian noise term. The outliers are then treated as unknowns and are estimated together with the model parameters. In the following sections, we first describe the regression model, followed by the Bayesian approach and the optimization approach.

### 2.1. Model Specification

Let $(\mathbf{x_i}, y_i), i = 1, 2, ..., N$ be the given training dataset with dependent variables $y_i, i = 1, 2, \ldots, N$ and independent variables $\mathbf{x_i}, i = 1, 2, \ldots, N$. In the RVM formulation, $y_i$ is related to $\mathbf{x_i}$ by the model

$$y_i = \sum_{j=1}^{N} w_j K(\mathbf{x_i}, \mathbf{x_j}) + w_0 + e_i \tag{1}$$

where with each $\mathbf{x_j}$, there is an associated kernel function $K(., \mathbf{x_j})$, and $e_i$ is the Gaussian noise. The objective is to estimate the weight vector $\mathbf{w} = [w_0, w_1, \ldots, w_N]^T$ using the training dataset. Once this is done, we can predict the dependent variable $y$ for any new $\mathbf{x}$ by

$$y = \sum_{i=j}^{N} w_j K(\mathbf{x}, \mathbf{x_j}) + w_0 \tag{2}$$

In the presence of outliers, Gaussian noise is not an appropriate assumption for $e_i$. We propose to split the noise $e_i$ into two components: a Gaussian component $n_i$ and a component due to outliers $s_i$, which we assume to be sparse. With this, we have

$$y_i = \sum_{j=1}^{N} w_j K(\mathbf{x_i}, \mathbf{x_j}) + w_0 + n_i + s_i \tag{3}$$

In matrix-vector form, this is given by

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \mathbf{n} + \mathbf{s} \tag{4}$$

where $\mathbf{y} = [y_1, \ldots, y_N]^T$, $\mathbf{n} = [n_1, \ldots, n_N]^T$, $\mathbf{s} = [s_1, \ldots, s_N]^T$ and $\mathbf{\Phi}$ is a $N \times (N + 1)$ matrix with $\mathbf{\Phi} = [\phi(\boldsymbol{x_1}), \phi(\boldsymbol{x_2}), \ldots, \phi(\boldsymbol{x_N})]^T$, where $\phi(\boldsymbol{x_i}) = [1, K(\mathbf{x_i}, \mathbf{x_1}), K(\mathbf{x_i}, \mathbf{x_2}), \ldots, K(\mathbf{x_i}, \mathbf{x_N})]^T$. The two unknowns $\mathbf{w}$ and $\mathbf{s}$ can be augmented into a single unknown vector $\mathbf{w_s} = [\mathbf{w}^T \mathbf{s}^T]^T$ and the above equation can be written as

$$\mathbf{y} = \mathbf{\Psi}\mathbf{w_s} + \mathbf{n} \tag{5}$$

where $\mathbf{\Psi} = [\mathbf{\Phi}|\mathbf{I}]$ is a $N \times (2N+1)$ matrix with $\mathbf{I}$, a $N \times N$ identity matrix.

### 2.2. Robust Bayesian RVM (RB-RVM)

In the Bayesian approach, we estimate the joint posterior distribution of $\mathbf{w}$ and $\mathbf{s}$, given the observations $\mathbf{y}$ and the prior distributions on $\mathbf{w}$ and $\mathbf{s}$. We then use the mean of the posterior distribution of $\mathbf{w}$ for prediction (2). The posterior variance also provides us with a measure of uncertainty in the prediction.

The joint posterior distribution of $\mathbf{w}$ and $\mathbf{s}$ is given by

$$p(\mathbf{w}, \mathbf{s}|\mathbf{y}) = \frac{p(\mathbf{w}, \mathbf{s})p(\mathbf{y}|\mathbf{w}, \mathbf{s})}{p(\mathbf{y})} \tag{6}$$

From (5), the likelihood term $p(\mathbf{y}|\mathbf{w}, \mathbf{s})$ is given by

$$p(\mathbf{y}|\mathbf{w}, \mathbf{s}) = \mathcal{N}(\mathbf{\Psi}\mathbf{w_s}, \sigma^2\mathbf{I}) \tag{7}$$

where $\sigma^2$ is the inlier Gaussian noise variance. To proceed further, we need to specify the prior distribution $p(\mathbf{w}, \mathbf{s})$. We assume that $\mathbf{w}$ and $\mathbf{s}$ are independent: $p(\mathbf{w}, \mathbf{s}) = p(\mathbf{w})p(\mathbf{s})$. Next, we keep the same 'sparsity promoting' prior for $\mathbf{w}$ as in RVM [17], that is,

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=0}^{N} \mathcal{N}(w_i|0, \alpha_i^{-1}) \tag{8}$$

where $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_N]^T$ is a vector of $(N + 1)$ hyper-parameters. A uniform distribution (hyper-prior) is assumed for each of the $\alpha_i$ (For more details, please see [17]).

For $\mathbf{s}$, we specify a similar sparsity promoting prior given by

$$p(\mathbf{s}|\boldsymbol{\beta}) = \prod_{i=1}^{N} \mathcal{N}(s_i|0, \beta_i^{-1}) \tag{9}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_N]^T$ is a vector of $N$ hyper-parameters, and each of the $\beta_i$ follows a uniform distribution. This completes the description of the likelihood $p(\mathbf{y}|\mathbf{w}, \mathbf{s})$ and the prior $p(\mathbf{w}, \mathbf{s})$. Next, we proceed to the inference stage.

### 2.2.1 Inference

Our inference method follows the RVM inference steps. We first find point-estimates for the hyper-parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and the inlier noise variance $\sigma^2$ by maximizing $p(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2)$ with respect to these parameters, where $p(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2)$ is given by

$$p(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2) = \int p(\mathbf{y}|\mathbf{w}, \mathbf{s}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})p(\mathbf{s}|\boldsymbol{\beta}) \, d\mathbf{w}d\mathbf{s}$$
(10)

Since all the distributions in the right hand side are Gaussian with zero mean, it can be shown that $p(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2)$ is a zero-mean Gaussian distribution with covariance matrix $\sigma^2\mathbf{I} + \boldsymbol{\Psi}\mathbf{A}^{-1}\boldsymbol{\Psi}^T$, where $\mathbf{A} = diag(\alpha_0, \ldots, \alpha_N, \beta_1, \ldots, \beta_N)$. The maximization of $p(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2)$ with respect to the hyper-parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and the noise variance $\sigma^2$ is known as evidence maximization and can be solved by an EM algorithm [17] or a faster implementation proposed in [18]. We will refer to these estimated parameters as $\boldsymbol{\alpha}_{MP}$, $\boldsymbol{\beta}_{MP}$ and $\sigma^2_{MP}$.

With this point estimation of the hyper-parameters and the noise variance, the (conditional) posterior distribution $p(\mathbf{w}, \mathbf{s}|\mathbf{y}, \boldsymbol{\alpha}_{MP}, \boldsymbol{\beta}_{MP}, \sigma^2_{MP})$ is given by

$$\frac{p(\mathbf{y}|\mathbf{w}, \mathbf{s}, \sigma^2_{MP})p(\mathbf{w}|\boldsymbol{\alpha}_{MP})p(\mathbf{s}|\boldsymbol{\beta}_{MP})}{p(\mathbf{y}|\boldsymbol{\alpha}_{MP}, \boldsymbol{\beta}_{MP}, \sigma^2_{MP})}$$
(11)

Since all the terms in the numerator are Gaussian, it can be shown that this is again a Gaussian distribution with covariance and mean given by

$$\boldsymbol{\Sigma} = (\sigma^{-2}_{MP}\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \mathbf{A_{MP}})^{-1} \text{ and } \boldsymbol{\mu} = \sigma^{-2}_{MP}\boldsymbol{\Sigma}\boldsymbol{\Psi}^T\mathbf{y} \quad (12)$$

where $\mathbf{A_{MP}} = diag(\alpha_{MP0}, \ldots, \alpha_{MPN}, \beta_{MP1}, \ldots, \beta_{MPN})$.

To obtain the posterior distribution $p(\mathbf{w}, \mathbf{s}|\mathbf{y})$, we need to integrate out $\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2$ from $p(\mathbf{w}, \mathbf{s}|\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2)$, that is,

$$p(\mathbf{w}, \mathbf{s}|\mathbf{y}) = \int p(\mathbf{w}, \mathbf{s}|\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2|\mathbf{y}) \, d\boldsymbol{\alpha}d\boldsymbol{\beta}d\sigma^2$$
(13)

However, this is analytically intractable; it has been empirically observed in [17], that for predictive purposes, $p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2|\mathbf{y})$ is very well approximated by $\delta(\boldsymbol{\alpha}_{MP}, \boldsymbol{\beta}_{MP}, \sigma^2_{MP})$. With this approximation, we have

$$p(\mathbf{w}, \mathbf{s}|\mathbf{y}) = p(\mathbf{w}, \mathbf{s}|\mathbf{y}, \boldsymbol{\alpha}_{MP}, \boldsymbol{\beta}_{MP}, \sigma^2_{MP}) \quad (14)$$

Thus, the desired joint posterior distribution of $\mathbf{w}$ and $\mathbf{s}$ is Gaussian with the posterior covariance and mean given by (12). This is the mean and covariance that we will use for prediction, as described next.

### 2.2.2 Prediction

We use the prediction model (2) to predict $\hat{y}$ for any new data $\hat{\mathbf{x}}$. The predictive distribution of $\hat{y}$ is given by

$$p(\hat{y}|\mathbf{y}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) = \int p(\hat{y}|\mathbf{w}, \sigma^2_{MP})p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha}_{MP}) \, d\mathbf{w}$$
(15)

where the posterior distribution of $\mathbf{w}$, $p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha}_{MP})$, can be easily obtained from the joint posterior distribution $p(\mathbf{w}, \mathbf{s}|\mathbf{y}, \boldsymbol{\alpha}_{MP}, \boldsymbol{\beta}_{MP}, \sigma^2_{MP})$. $p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha}_{MP})$ is a Gaussian distribution with mean and covariance given by the mean and covariance of the parameter part ($\mathbf{w}$) of the $\mathbf{w_s}$ vector, that is,

$$\boldsymbol{\Sigma_w} = \boldsymbol{\Sigma}(1:N+1, 1:N+1) \text{ and } \boldsymbol{\mu_w} = \boldsymbol{\mu}(1:N+1)$$
(16)

With this, it can be shown that the predictive distribution of $\hat{y}$ is Gaussian with mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ given by

$$\hat{\mu} = \boldsymbol{\mu_w}^T\boldsymbol{\phi}(\hat{\mathbf{x}}) \text{ and } \hat{\sigma}^2 = \sigma^2_{MP} + \boldsymbol{\phi}(\hat{\mathbf{x}})^T\boldsymbol{\Sigma_w}\boldsymbol{\phi}(\hat{\mathbf{x}}) \quad (17)$$

### 2.2.3 Advantage over other Robust RVM Algorithms

The proposed robust Bayesian formulation (RB-RVM) is very similar to the original RVM formulation. All we have to do is, instead of inferring just the parameter vector $\mathbf{w}$, infer the joint parameter-outlier vector $\mathbf{w_s}$, by replacing the $\boldsymbol{\Phi}$ matrix with the corresponding $\boldsymbol{\Psi} = [\boldsymbol{\Phi}|\mathbf{I}]$ matrix, and use only the parameter part of the estimated $\mathbf{w_s}$ for prediction. It is this simple modification of the original RVM that gives RB-RVM the computational advantage over [6, 19, 22] because we can use the fast implementation of RVM [18] to solve the robust RVM problem.

### 2.3. Basis Pursuit RVM (BP-RVM)

A very similar objective, as in the Bayesian approach, can be achieved by solving the following optimization problem:

$$\min_{\mathbf{w_s}} ||\mathbf{w_s}||_0 \text{ subject to } ||\mathbf{y} - \boldsymbol{\Psi}\mathbf{w_s}||_2 \le \epsilon \quad (18)$$

where $||\mathbf{w_s}||_0$ is the $L_0$ norm, which counts the number of non-zero elements in $\mathbf{w_s}$. The cost function promotes a sparse solution for $\mathbf{w_s}$ and the constraint term is essentially the likelihood term of the Bayesian approach, with $\epsilon$ related to the inlier noise variance $\sigma^2$. $\mathbf{w}$ obtained after solving this problem can be used for prediction. However, this is a combinatorial problem; hence, it cannot be solved directly. This problem has been studied extensively in sparse representation literature [4, 5]. In one of the approaches, a convex relaxation of the problem is solved

$$\min_{\mathbf{w_s}} ||\mathbf{w_s}||_1 \text{ subject to } ||\mathbf{y} - \boldsymbol{\Psi}\mathbf{w_s}||_2 \le \epsilon \quad (19)$$

where the $L_0$ norm in the cost function is replaced by the $L_1$ norm, which makes it a convex problem; hence, it can be solved in polynomial time. This approach is known as Basis Pursuit Denoising (BPD) [4, 5], and we will refer to the robust algorithm based on BPD as the Basis Pursuit RVM (BP-RVM). Initially, the justification for using the $L_1$ norm approximation was based on empirical observations [4]. However, recently in [3, 5], it has been shown

that if $\mathbf{w_s}$ was sparse to begin with, then under certain condition ('Restricted Isometry Property' or 'incoherence') on the matrix $\mathbf{\Psi}$, (18) and (19) will have the same solution up to a bounded uncertainty due to $\epsilon$. However, in our case the matrix $\mathbf{\Psi}$ depends on the training dataset and the associated kernel function, and it might not satisfy the desired conditions mentioned above.
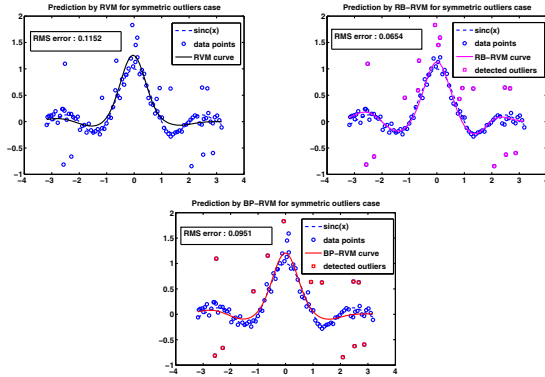


Figure 1. Prediction by the three algorithms: RVM, RB-RVM and BP-RVM in the presence of symmetric outliers for $N = 100$, $f = 0.2$ and $\sigma = 0.1$. Data which are enclosed by a box are the outliers found by the robust algorithms. Prediction error are also shown in the figures. RB-RVM gives the lowest prediction error.
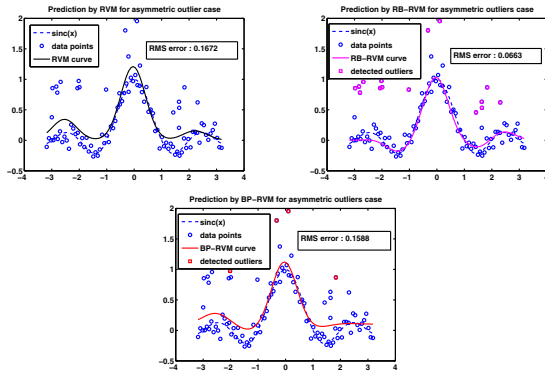


Figure 2. Prediction by the three algorithms: RVM, RB-RVM and BP-RVM in the presence of asymmetric outliers for $N = 100$, $f = 0.2$ and $\sigma = 0.1$. Data which are enclosed by a box are the outliers found by the robust algorithms. Prediction error are also shown in the figures. Clearly, RB-RVM gives the best result.

## 3. Empirical Evaluation

In this section, we empirically evaluate the proposed robust versions of the RVM, RB-RVM and BP-RVM, with respect to the baseline RVM. We consider three important intrinsic parameters of the robust regression problem: the outlier fraction ($f$), the inlier noise variance ($\sigma^2$) and the number of training data points ($N$), and study the performance of the three algorithms for different settings of these parameters.[1] Next, we describe the experimental setup, which is

[1]For solving RVM and RB-RVM, we have used the publicly available code in http://www.vectoranomaly.com/downloads/downloads.htm.

quite similar to that of [6].

We generate our training data using the normalized sinc function $sinc(x) = sin(\pi x)/(\pi x)$. $y_i$ of the inlier data are obtained by adding a Gaussian noise $\mathcal{N}(0, \sigma^2)$ to $sinc(x_i)$. For the outliers, we consider two generative models: 1) symmetric and 2) asymmetric. In the symmetric model, $y_i$ is obtained by adding a uniform noise of range $[-1, +1]$ to $sinc(x_i)$, and in the asymmetric model, $y_i$ is obtained by adding a uniform noise of range $[0, +1]$ to $sinc(x_i)$. With each training data $x_j$, we associate a Gaussian kernel: $K(x, x_j) = \exp\left(-(x - x_j)^2/r^2\right)$, with $r = 2$. Figure 1 and 2 show the performance of the three algorithms for the symmetric and asymmetric outlier cases for $N = 100$, $f = 0.2$ and $\sigma = 0.1$. The performance criteria used for comparison is the root mean square (RMS) prediction error. Note that, after inference, robust methods can also classify the training data as inliers or outliers. We classify a data as an outlier if the prediction error (absolute difference between the predicted and the observed value) is greater than three times the inlier noise standard deviation, which is also estimated during inference. From figure 1 and 2, we conclude that RB-RVM gives the lowest prediction error, followed by BP-RVM and RVM. In the following sections, we study the performance of the algorithms by varying the intrinsic parameters: $f$, $\sigma$ and $N$.

**Varying the Outlier fraction:** We vary the outlier fraction $f$, with the other parameters fixed at $N = 100$ and $\sigma = 0.1$. Figure 3 shows the prediction error vs. outlier fraction for the symmetric and asymmetric outliers cases. For both the cases, RB-RVM gives the best result. For the symmetric case, BP-RVM gives lower prediction error than RVM but for the asymmetric case they give similar result.
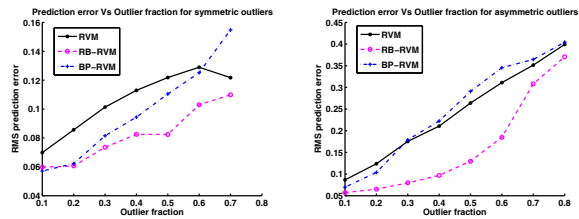


Figure 3. Prediction error vs. outlier fraction for the symmetric and asymmetric outlier cases. RB-RVM gives the best result for both the cases. For the symmetric case, BP-RVM gives lower prediction error than RVM but for the asymmetric case they give similar result.

**Varying the Inlier Noise Std:** We vary the inlier noise standard deviation $\sigma$, with the other parameters fixed at $N = 100$ and $f = 0.2$. Figure 4 shows that RB-RVM gives the lowest prediction error until about $\sigma = 0.2$, after which RVM gives better result. This is because for our experimental setup, at approximately $\sigma = 0.3$, the distinction between the inliers and outliers cease to exist. For Gaussian distribution, most of the probability density mass lies

For solving BP-RVM, we have used l1-magic: http://www.acm.caltech.edu/l1magic/

within $3\sigma$ of the mean, and any data within this region can be considered as inliers and those outside as outliers. Thus, for $\sigma = 0.3$, $3\sigma = 0.9$; most of the outliers will be within this range and effectively become inliers.
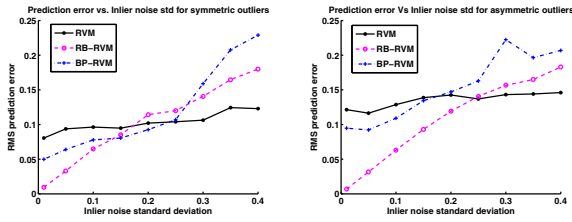


Figure 4. Prediction error vs. inlier noise standard deviation for the symmetric and asymmetric outlier cases. RB-RVM gives the lowest prediction error until about $\sigma = 0.2$, after which RVM gives better result. This is because for our experimental setup, at approximately $\sigma = 0.3$, the distinction between the inliers and outliers cease to exist.

**Varying the Number of Data Points:** We vary the number of data points $N$, with $f = 0.2$ and $\sigma = 0.1$. Figure 5 shows that the performance of all the three algorithms improve with increasing $N$.
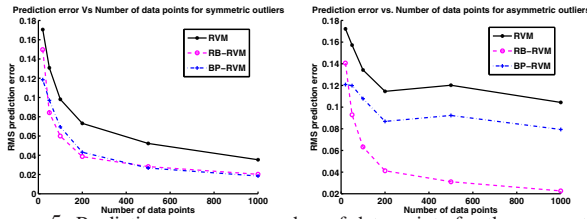


Figure 5. Prediction error vs. number of data points for the symmetric and asymmetric outlier cases. For all the three algorithms, performance improves with increasing $N$.

**Discussion:** From the above study, we conclude that in presence of outliers RB-RVM and BP-RVM perform better than RVM. The performance of BP-RVM is poor as compared to RB-RVM; this indicates that the $L_1$ norm relaxation (19) is not a good approximation of the $L_0$ norm problem (18), when $\Psi$ does not satisfy the desired Restricted Isometry Property [3]. Henceforth, we will only consider RB-RVM for solving the image denoising and age regression problems.

## 4. Robust Image Denoising

Recently, kernel regression has been used for solving a number of traditional image processing tasks such as image denoising, image interpolation and super-resolution with a great deal of success [15, 16]. The success of these kernel regression methods prompted us to test RB-RVM for solving the problem of image denoising in the presence of salt and pepper noise. Salt and pepper noise are randomly occurring white and black pixels in an image and can be considered as outliers.

Any image $I(x, y)$ can be considered as a surface over a 2D grid. Given a noisy image, we can use regression to learn the relation between the intensity and the 2D grid of
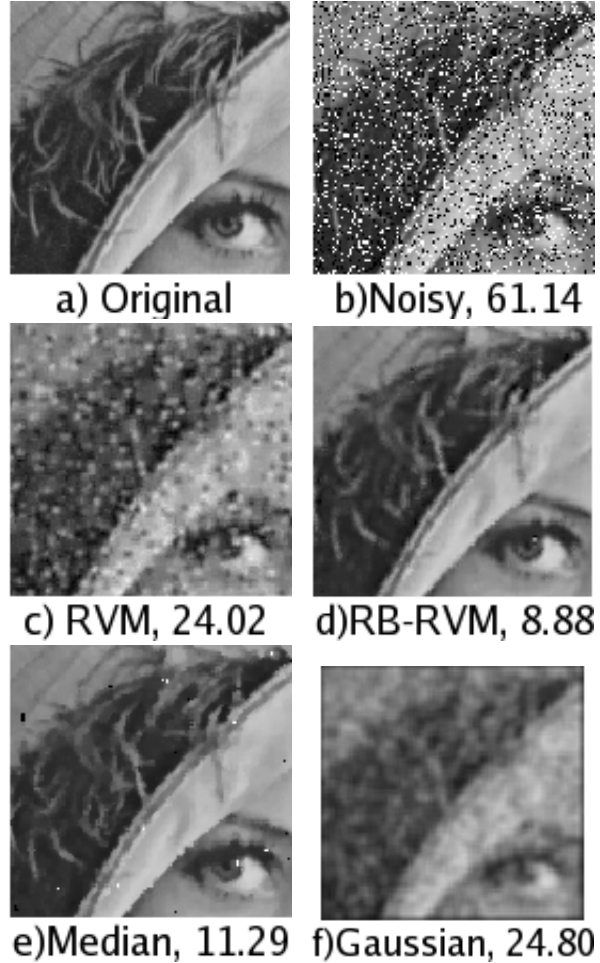


Figure 6. Salt and pepper noise removal experiment: the figure shows original image, noisy image and denoised images by RVM, RB-RVM, median filter and Gaussian filter. The corresponding RMSE values are also shown in the figure. Clearly, RB-RVM gives the best denoising result.

the image. If some kind of a local smoothness is imposed by the regression machine, we can use it for denoising the image. Here, we consider RVM and RB-RVM for achieving this purpose. We divide the image into many (overlapping) patches, and for each patch we infer the parameters of RVM and RB-RVM. We then use the inferred parameters for predicting the intensity of the central pixel of the patch, which is the denoised intensity at that pixel. This is done for all the pixels of the image to obtain the denoised image. Motivated by [15], we consider a composition of Gaussian and polynomial kernel as the choice of kernel in our regression machines. The Gaussian kernel is defined as $K_g(\mathbf{x}, \mathbf{x_j}) = \exp\left(-\|\mathbf{x} - \mathbf{x_j}\|^2/r^2\right)$, where $r$ is the scale of the Gaussian kernel, and the polynomial kernel is defined as $K_p(x, x_j) = (\mathbf{x}^T\mathbf{x_j} + 1)^p$, where $p$ is the order of the polynomial kernel. We consider kernels of the form: $K(\mathbf{x}, \mathbf{x_j}) = K_g(\mathbf{x}, \mathbf{x_j})K_p(\mathbf{x}, \mathbf{x_j})$.

To test the proposed kernel denoising algorithms, we follow the experimental setup of [15]: we add $20\%$ salt and

| RB-RVM | Wavelet [12] | $l_2$ Classic [15] | $l_2$ steering [15] | $l_1$ steering [15] |
|--------|--------------|--------------------|---------------------|---------------------|
| 9.24   | 21.54        | 21.81              | 21.06               | 7.14                |

Table 1. RMSE values for RB-RVM, Wavelet, $l_2$ Classic, $l_2$ steering, and $l_1$ steering. RB-RVM is better than all the algorithms except for the $l_1$ steering kernel regression.
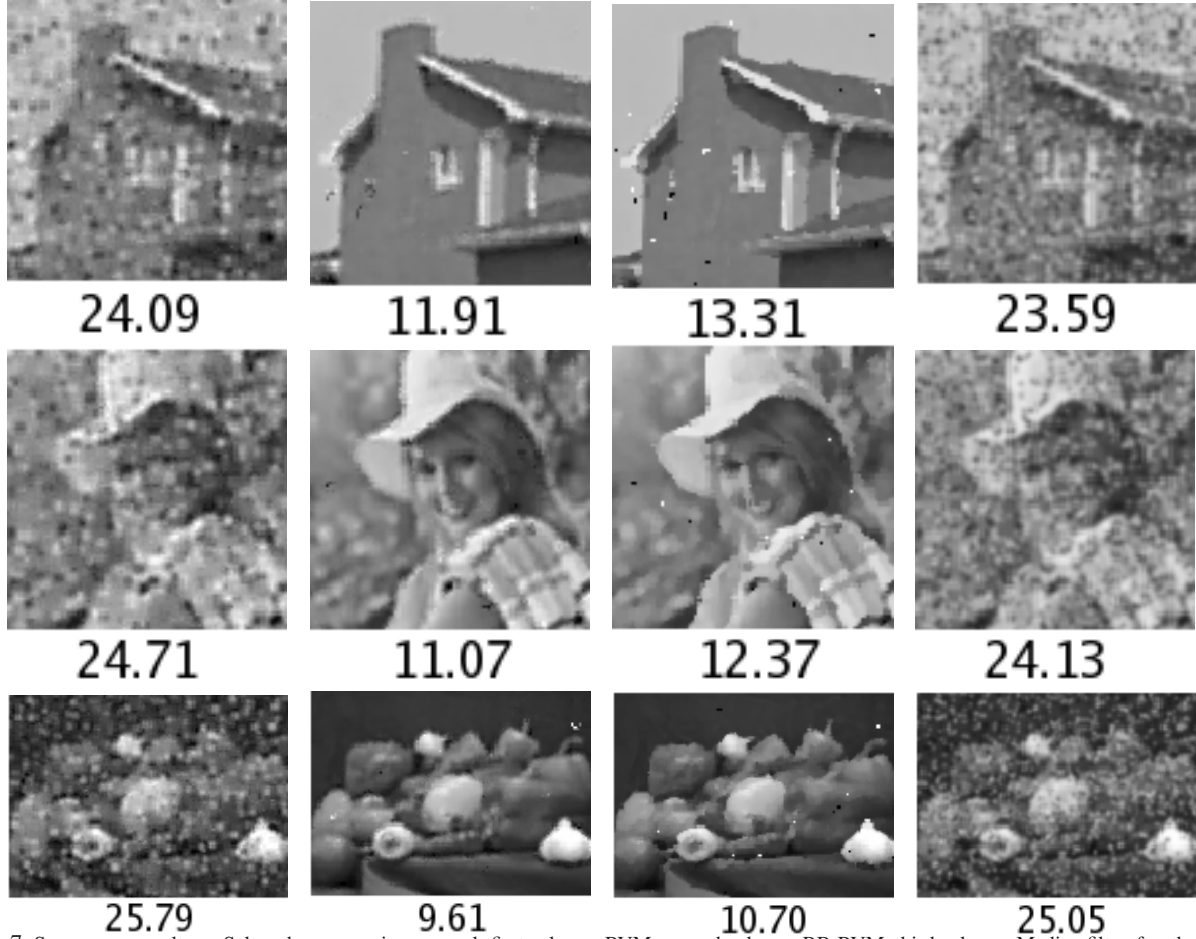


Figure 7. Some more results on Salt and pepper noise removal: first column: RVM, second column: RB-RVM, third column: Median filter, fourth column: Gaussian filter. The RMSE values are also shown in the figure; RB-RVM gives the best result.

pepper noise to the original image shown in figure 6. For RVM and RB-RVM, we choose patch size of $6 \times 6$, $r = 2.1$ and $p = 1$. Figure 6 shows the image denoising result by RVM, RB-RVM, the $3 \times 3$ median filter and the Gaussian filter (standard deviation $= 2.1$). The denoised images and the corresponding RMSE values show that RB-RVM gives the best denoising result. Table 1 further compares RB-RVM with other kernel regression algorithms (RMSE values taken from [15]), from which we conclude that RB-RVM is better than all the algorithms except for the $l_1$ steering kernel regression. Figure 7 shows some more denoising results. Next, we vary the amount of salt and pepper noise, and obtain the mean RMSE value over seven commonly used images of Lena, Barbara, House, Boat, Baboon, Pepper and Elaine. Figure 8 shows that RB-RVM gives better result than the median filter, which is the most commonly used filter for denoising images with salt and pepper noise.

Further, we test RB-RVM for denoising an image corrupted by a mixture of Gaussian noise of $\sigma = 5$ and 5% salt and pepper noise. From figure 9, we conclude again that RB-RVM gives much better denoising result as compared to RVM.
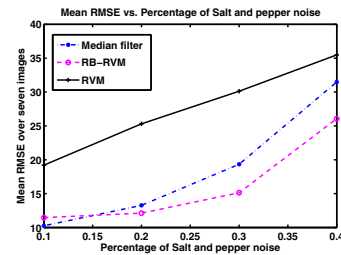


Figure 8. Mean RMSE value over seven images vs. percentage of salt and pepper noise. RB-RVM gives better performance than the median filter.
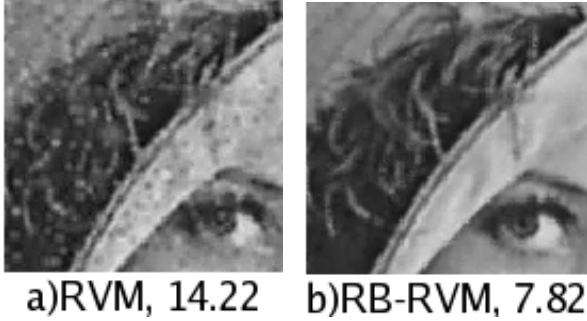
Figure 9. Mixture of Gaussian and salt and pepper noise removal experiment: denoised images by RVM and RB-RVM with their corresponding RMSE values. This experiment again shows that the RB-RVM based denoising algorithm gives much better result than the RVM based one.

## 5. Age Estimation from Facial Images

The goal of facial age estimation is to estimate the age of a person from his/her image. The most common approach for solving this problem is to extract some relevant features from the image, and then learn the functional relationship between these features and the age of the person using regression techniques [10, 9, 7, 8]. Here, we intend to test the RB-RVM regression for the age estimation problem. For our experiments, we use the publicly available FG-Net dataset [1], which contains 1002 images of 82 subjects at different ages. As a choice of features, we use geometric features proposed in [20], which are obtained by computing the 'flow field' at 68 fiducial points with respect to a reference face image.

To decide on a particular kernel for regression, we perform leave-one-person-out testing, by RB-RVM, for different choices of kernel. Table 2 shows the mean absolute error (MAE) of age prediction for different values of the scale parameter $r$ of the Gaussian kernel. $r = 0.2$ gives the best result, and we use this value of $r$ for all the subsequent experiments. Next, we use RB-RVM to categorize the whole dataset into inliers and outliers. The algorithm found 90 outliers; some of the inliers and outliers are shown in figure 10. With this knowledge of the inliers and the outliers, we perform the leave-one-person-out test again. Table 3 shows the mean absolute error (MAE) of age prediction for the inliers and the outliers separately. The small prediction error for the inliers and the large prediction error for the outliers indicate that the inlier vs. outlier categorization by RB-RVM was good. Table 3 also shows that the prediction error of the RB-RVM for the whole dataset is lower than that of the RVM. To put the numbers in the table in context, the state-of-the-art algorithm [8] gives a prediction error of 5.07 as compared to the prediction error of 4.61 obtained for the inliers by the RB-RVM.

To further test RB-RVM, we add various amount of controlled outliers. Before doing this, we remove the outliers found in the previous experiment. We use 90% of this new dataset as the training set and the remaining 10% as the test

| r | 0.1 | 0.2 | 0.3 | 0.4 |
|---|-----|-----|-----|-----|
| MAE | 7.10 | 6.52 | 6.54 | 6.62 |

Table 2. Mean absolute error (MAE) of age prediction for different values of the scale parameter $r$ of the Gaussian kernel. The prediction errors are for the leave-one-person-out testing by RB-RVM. $r = 0.2$ gives the best result, and we use this $r$ for all the subsequent experiments.



Figure 10. Some inliers and outliers found by RB-RVM. Most of the outliers are images of older subjects like Outlier A and B. This is because there are less number of samples of older subjects in the FG-Net database. Outlier C has an extreme pose variation from the usual frontal faces of the database; hence, it is an outlier. The facial geometry of Outlier D is very similar to that of younger subjects, such as big forehead and small chin, so it is classified as an outlier.

|  | Inlier MAE | Outlier MAE | All MAE |
|---|-----|-----|-----|
| RB-RVM | 4.61 | 25.87 | 6.52 |
| RVM | N.A. | N.A. | 6.80 |

Table 3. Mean absolute error (MAE) of age prediction for the inliers, outliers and the whole dataset using RB-RVM. Since RVM does not differentiate between inliers and outliers, we only show the prediction error for the whole dataset. The small MAE for the inliers and the large MAE for the outliers indicates that the inlier vs. outlier categorization by RB-RVM was good. Also, note that the prediction error of the RB-RVM for the whole dataset is lower than that of the RVM.
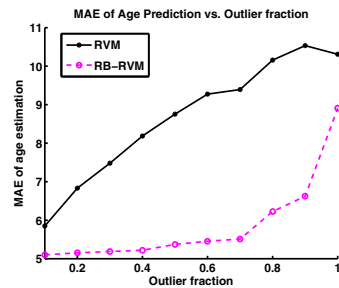


Figure 11. Mean absolute error (MAE) of age prediction vs. fraction of controlled outliers added to the training dataset. RB-RVM gives much lower prediction error as compared to the RVM. Also, note that the prediction error is reasonable even with outlier fraction as high as 0.7.

set. We introduce controlled outliers only in the training set, and perform age prediction on the test set by both RVM and RB-RVM. We vary the fraction of the outliers on the training set and measure the age prediction error on the test set. Figure 11 shows that RB-RVM gives much lower prediction error as compared to RVM. This experiment again suggests that RB-RVM should be preferred over RVM for

the age estimation problem.

## 6. Discussion and Conclusion

We explored two natural approaches for incorporating robustness to the RVM regression: a Bayesian approach and an optimization approach. In the Bayesian approach (RB-RVM), the robust RVM problem is formulated as a bigger RVM problem with the advantage that it can be solved efficiently by a fast algorithm. The optimization approach (BP-RVM) is based on the Basis Pursuit Denoising algorithm, which is a popular algorithm in the sparse representation literature. Empirical evaluations of the two robust algorithms show that RB-RVM performs better than BP-RVM. Further, we used RB-RVM to solve the robust image denoising and age estimation problem, which clearly demonstrated the superiority of RB-RVM over the original RVM.

## References

[1] The fg-net aging database, http://www.fgnet.rsunit.com. 7

[2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE TPAMI*, 2006. 1

[3] E. J. Candes and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 2008. 3, 5

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Jour. Scient. Comp.*, 1998. 1, 3

[5] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1), 2006. 3

[6] A. C. Faul and M. E. Tipping. A variational approach to robust regression. In *ICANN*, 2001. 1, 2, 3, 4

[7] Y. Fu, Y. Xu, and T. S. Huang. Estimating human age by manifold analysis of face pictures and regression on aging features. In *ICME*, 2007. 1, 7

[8] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang. Image-based human age estimation by manifold learning and locally adjusted robust regression. *IEEE Transactions on Image Processing*, 2008. 1, 7

[9] A. Lanitis, C. Draganova, and C. Christodoulou. Comparing different classifiers for automatic age estimation. *IEEE TSMC*, 2004. 1, 7

[10] A. Lanitis, C. J. Taylor, and T. F. Cootes. Toward automatic simulation of aging effects on face images. *IEEE TPAMI*, 2002. 1, 7

[11] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE TPAMI*, 31, 2009. 1

[12] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 2003. 6

[13] C. E. Rasmussen and C. K. I. Williams. Gaussian processes for machine learning. *The MIT press*, 2006. 1

[14] T. Sim and T. Kanade. Combining models and exemplars for face recognition: An illuminating example. In *CVPR 2001 Workshop on Models versus Exemplars in Computer Vision*, 2001. 1

[15] H. Takeda, S. Farsiu, and P. Milanfar. Robust kernel regression for restoration and reconstruction of images from sparse noisy data. In *ICIP*, 2006. 1, 5, 6

[16] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE TIP*, 2007. 1, 5

[17] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 2001. 1, 2, 3

[18] M. E. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003. 1, 3

[19] M. E. Tipping and N. D. Lawrence. Variational inference for student- models: Robust bayesian interpolation and generalised component analysis. *Neurocomputing*, 69(1-3), 2005. 1, 2, 3

[20] P. Turaga, S. Biswas, and R. Chellappa. Role of geometry of age estimation. In *ICASSP*, 2010. 7

[21] V. N. Vapnik. The nature of statistical learning theory. 1995. 1

[22] B. Yang, Z. Zhang, and Z. Sun. Robust relevance vector regression with trimmed likelihood function. In *IEEE Sig. Proc. Letters*, 2007. 1, 2, 3