Bayesian Regression with Input Noise for High Dimensional Data

Jo-Anne Ting JOANNETI@USC.EDU
University of Southern California, Los Angeles, CA 90089

Aaron D'Souza ADSOUZA@GOOGLE.COM
Google, Inc., Mountain View, CA 94043

Stefan Schaal SSCHAAL@USC.EDU
University of Southern California, Los Angeles, CA 90089 and ATR Computational Neuroscience Laboratories, Kyoto, Japan

Abstract

This paper examines high dimensional regression with noise-contaminated input and output data. Goals of such learning problems include optimal prediction with noiseless query points and optimal system identification. As a first step, we focus on linear regression methods, since these can be easily cast into nonlinear learning problems with locally weighted learning approaches. Standard linear regression algorithms generate biased regression estimates if input noise is present and suffer numerically when the data contains redundancy and irrelevancy. Inspired by Factor Analysis Regression, we develop a variational Bayesian algorithm that is robust to ill-conditioned data, automatically detects relevant features, and identifies input and output noise – all in a computationally efficient way. We demonstrate the effectiveness of our techniques on synthetic data and on a system identification task for a rigid body dynamics model of a robotic vision head. Our algorithm performs 10 to 70% better than previously suggested methods.

1. Introduction

Learning the equations of motion of a complex physical system for the purpose of control is a common problem in robotics. A typical system identification approach would first collect a representative data set from the robot by measuring positions and motor commands during some explorative movements. Then, velocity and acceleration information would be obtained by numerical differentiation of position data. The data would also be digitally filtered to reduce noise. As a third step, a function approximator would be applied to learn the mapping from positions, velocities and accelerations to motor commands. Such a function often has hundreds of inputs for complex robots. Finally, this mapping would be inserted into the control loop of the robot, where appropriate motor commands are predicted from desired position, velocity and acceleration information – all of which are noiseless data.

The example scenario above is representative for a large number of system identification problems. From a machine learning point of view, the interesting components are that the learning data is high dimensional, has irrelevant and redundant dimensions and, despite digital filtering, usually contains a significant amount of noise in the inputs to the function approximator. Moreover, predictions are required from *noiseless* input data, since inputs generated during control originate from a planning system without noise. The quality of control strongly depends on the quality of the learned internal model in advanced controllers and is critical in many robotic applications such as haptic devices, surgical robotics and safe compliant assistive robots in human environments.

In this paper, we address the problem above in the context of linear regression, since an extension to nonlinear regression is straightforward using Locally Weighted Learning methods [Atkeson et al., 1997]. If we wanted to use traditional linear regression techniques in the sample application given, we would encounter several deficiencies. First, algorithms like Ordinary Least Squares (OLS) regression do not address noise in the input data and will result in regression solutions with a persistent bias. Alternative methods such as Total Least Squares (TLS) [Golub &

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

Van Loan, 1989,Van Huffel & Vanderwalle, 1991] (otherwise known as orthogonal-least squares regression [Hollerbach & Wampler, 1996]) address input noise, but they assume the variances of input noise and output noise are the same [Rao & Principe, 2002]. In real-world systems, this assumption is not necessarily true and, again, the resulting estimates will be biased, leading to inferior generalization. Additionally, for ill-conditioned data in high dimensional spaces, most traditional linear regression techniques break down numerically since they are unable to generate sparse solutions identifying redundant and/or relevant dimensions. Algorithms such as stepwise regression [Draper & Smith, 1981]and LASSO (Least Absolute Shrinkage and Selection Operator) regression [Tibshirani, 1996] have been suggested to produce sparse solutions. Unfortunately, they ignore the effect of noise in input data and require careful human supervision to ensure useful results.

This paper is structured as follows. First, we motivate the problem of input noise in linear regression applications and identify possible solutions. Then, we introduce a novel technique that incorporates input noise detection and uses Bayesian regularization methods to ensure robustness to ill-conditioned data. Finally, we evaluate our approach on synthetic data and on a 7 degree-of-freedom (DOF) robotic vision head.

2. High Dimensional Regression with Input Noise

Let us examine some of the problems associated with traditional linear regression methods before introducing our de-noising solution. Assuming that the input vectors $\mathbf{x}$ are arranged in the rows of the matrix $\mathbf{X}$ and the corresponding scalar outputs y are the coefficients of the vector $\mathbf{y}$, the general model for linear regression with noise-contaminated input and output data can be expressed as follows:

$$y = \sum_{m=1}^{d} w_{zm} t_m + \epsilon_y \qquad x_m = w_{xm} t_m + \epsilon_{xm} \quad (1)$$

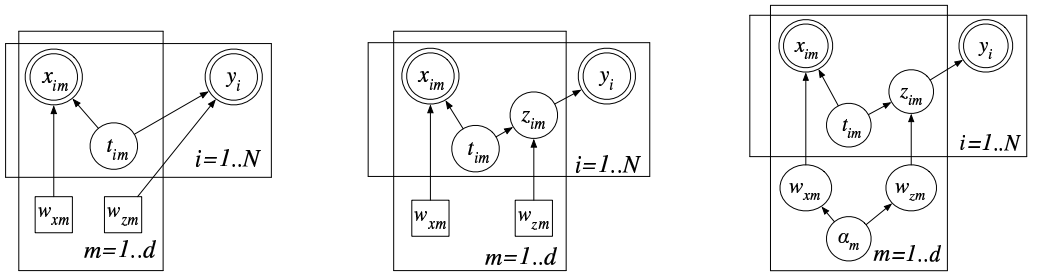
where d is the number of input dimensions, $\mathbf{t}$ is noiseless input data composed of t_m components, $\mathbf{w_z}$ and $\mathbf{w_x}$ are regression vectors composed of w_{zm} and w_{xm} components respectively, and ϵ_y and ϵ_x are additive mean-zero noise. Only $\mathbf{X}$ and y are observable. Note that if the input data is noiseless (that is, $x_m = w_{xm} t_m$), then we obtain the familiar linear regression equation of $y = \beta_{OLS}^T \mathbf{x} + \epsilon_y$, where $\beta_{OLS,m} = w_{zm}/w_{xm}$. The slightly more general formulation with the distinct coefficients w_{xm} and w_{zm} used above will be useful in preparing our new algorithm.

The OLS estimate of the regression vector β_{OLS} is $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. The first major issue with OLS regression in high dimensional spaces is that the full rank assumption of $(\mathbf{X}^T\mathbf{X})^{-1}$ is often violated due to underconstrained data sets. For more than 500 input dimensions, the matrix inversion required in OLS also becomes rather expensive. Ridge regression can fix the problem of ill-conditioned matrices by introducing an uncontrolled amount of bias. There exist also alternative methods to invert the matrix more efficiently [Strassen, 1969,Hastie & Tibshirani, 1990], as for instance through singular value decomposition factorization. Nevertheless, all these methods are unable to model noise in input data and require manual tuning of meta parameters, which can strongly influence the quality of the estimation results.

If we examine Eq. (1), it can be shown that the OLS estimate in the presence of noise will be $\beta_{OLS,noise} = \gamma\beta_{true}$, where $0 < \gamma < 1$, and its exact value depends on the amount of input noise. Thus, OLS regression underestimates the true regression vector β_{true} and generates biased predictions, a problem that cannot be fixed by adding more training data.

Intentionally, the input/output noise model formulation in Eq. (1) was chosen such that it coincides with a version of Factor Analysis [Massey, 1965] tailored for regression problems. The intuition of this model is given in Figure 1(a): every observed input x_{im} and output y_i is assumed to be generated by a set of hidden variables t_{im} and contaminated with some noise, exactly as given in Eq. (1). The graphical model in Figure 1(a) compactly describes the full multi-dimensional system. The goal of learning is to find parameters w_{xm} and w_{zm}, which can only be achieved by estimating the hidden variables t_{im}, z_{im} and the variances of all random variables. With this knowledge, optimal prediction can be performed with either noisy or noiseless inputs, by deriving the appropriate conditional distributions (see below). The specific version of factor analysis for regression depicted in Figure 1(a) is called joint-space Factor Analysis or Joint Factor Analysis (JFA), since both input and output variables are actually treated in the same way during the estimation process, i.e., only their joint distribution matters. While Joint Factor Analysis is well suited for modeling regression models with input noise, it does not handle ill-conditioned data very well and is computationally prohibitive in high dimensions.

In the following section, we will develop a Bayesian treatment of Joint Factor Analysis that is robust to ill-conditioned data, automatically detects non-identifiable parameters, detects noise in input and out-



(a) Joint Factor Analysis (b) A modified model of JFA for efficient estimation (c) A Bayesian version of JFA

Figure 1. Graphical Models for Noisy Linear Regression. Random variables are in circular nodes, observed random variables are in double circles, and point estimated parameters are in square nodes. d is the total number of input dimensions while N is the total number of samples in the data set.

put data and does this all in a computationally inexpensive manner.

3. Bayesian Parameter Estimation of Noisy Linear Regression

Figure 1 illustrates the progression of modifications made to the graphical model of Joint Factor Analysis in order to derive our Bayesian version.

3.1. EM-based Joint Factor Analysis

To start with, we introduce the hidden variables z_{im} such that $z_{im} = w_{zm}t_{im}$. This trick [D'Souza et al., 2004] allows us to avoid any form of matrix inversion in the resulting learning algorithm. Our noisy linear regression model from Eq. (1) thus becomes:

$$\begin{aligned} y_i &= \sum_{m=1}^{d} z_{im} + \epsilon_y \\ z_{im} &= w_{zm}t_{im},\ x_m = w_{xm}t_m + \epsilon_{xm} \end{aligned} \tag{2}$$

To determine all open parameters in the context of maximum likelihood estimation, we use the Expectation-Maximization (EM) algorithm [Dempster et al., 1977], and the following standard assumptions about the underlying probability distributions are made:

$$\begin{aligned} y_i &\sim \text{Normal}(\mathbf{1}^T z_i, \psi_y) & z_{im} &\sim \text{Normal}(w_{zm}t_{im}, \psi_{zm}) \\ x_{im} &\sim \text{Normal}(w_{xm}t_{im}, \psi_{xm}) & t_{im} &\sim \text{Normal}(0,1) \end{aligned}$$

where $\mathbf{1} = [1,1,...1]^T$, $\mathbf{z_i}$ is a d by 1 vector, $\mathbf{w_z}$ is a d by 1 vector composed of w_{zm} elements, and $\mathbf{w_x}$, $\psi_\mathbf{z}$ and $\psi_\mathbf{x}$ are similarly composed of w_{xm}, ψ_{zm} and ψ_{zm} elements, respectively. As Figure 1(b) shows, the regression coefficients w_{zm} are now behind the fan-in to the output y_i. This new formulation of Joint Factor Analysis decouples the input dimensions and generates a learning algorithm that operates with $O(d)$ computational complexity per EM iteration, where d is the number of input dimensions, instead of approximately $O(d^3)$ as in traditional Joint Factor Analysis.

3.2. Automatic Feature Detection

The efficient maximum likelihood formulation of Joint Factor Analysis is, however, still vulnerable to ill-conditioned data. Thus, we introduce a Bayesian layer on top of this model by treating the regression parameters $\mathbf{w_z}$ and $\mathbf{w_x}$ probabilistically to protect against overfitting, as shown in Figure 1(c). To do this, we introduce "precision" variables α_m over each regression parameter w_{zm}. The same α_m is also used for each w_{xm}, leading to a coupled regularization of w_{zm} and w_{xm}. As a result, the regression parameters are now distributed as follows: $w_{zm} \sim \text{Normal}(0, 1/\alpha_m)$ and $w_{xm} \sim \text{Normal}(0, 1/\alpha_m)$, where α_m takes on a Gamma distribution, shown below:

$$\begin{aligned} p(\mathbf{w_z}|\alpha) &= \prod_{m=1}^{d} \left(\frac{\alpha_m}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{\alpha_m}{2} w_{zm}^2\right\} \\ p(\mathbf{w_x}|\alpha) &= \prod_{m=1}^{d} \left(\frac{\alpha_m}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{\alpha_m}{2} w_{xm}^2\right\} \\ p(\alpha) &= \prod_{m=1}^{d} \frac{b_{\alpha_m}^{a_{\alpha_m}}}{\Gamma(a_{\alpha_m})} \alpha_m^{(a_{\alpha_m}-1)} \exp\{-b_{\alpha_m}\alpha_m\} \end{aligned} \tag{3}$$

The rationale of this Bayesian modeling technique is as follows. The key quantity that determines the relevance of a regression input is the parameter α_m. A priori, we assume that every w_m has a mean zero distribution with broad variance $1/\alpha_m$. If the posterior value of α_m turns out to be very large after all model parameters are estimated, then the corresponding posterior distribution of w_{zm} must be sharply peaked at zero. Thus, this gives strong evidence that $w_{zm} = 0$ and that the input t_m contributes no information to the regression model. If an input t_m contributes no information to the output, then it is irrelevant to the regression and it is also irrelevant how much it contributes to x_m. That is to say, the corresponding inputs x_m could be treated as pure noise. Coupling both w_{zm} and w_{xm} with the same precision variable α_m accomplishes exactly this effect. Thus, the Bayesian approach automatically detects irrelevant input dimen-

sions and regularizes against ill-conditioned data sets, while detecting noise in input and output data.

Even with the Bayesian layer added, the entire regression problem can be treated as an EM-like learning problem [Ghahramani & Beal, 2000]. Given the data $D = \{\mathbf{x}_i, y_i\}_{i=1}^N$, we maximize the incomplete log likelihood $\log p(\mathbf{y}|\mathbf{X})$ by maximizing the expected complete log likelihood $\langle \log p(\mathbf{y}, \mathbf{Z}, \mathbf{T}, \mathbf{w}_z, \mathbf{w}_x, \alpha, |\mathbf{X}) \rangle$, where:

$$\log p(\mathbf{y}, \mathbf{Z}, \mathbf{T}, \mathbf{w}_z, \mathbf{w}_x, \alpha|\mathbf{X}) = -\frac{N}{2}\log\psi_y - \frac{1}{2\psi_y}\sum_{i=1}^{N}\left(y_i - \mathbf{1}^T z_i\right)^2$$
$$- \frac{N}{2}\sum_{m=1}^{d}\log\psi_{zm} - \sum_{m=1}^{d}\frac{1}{\psi_{zm}}\sum_{i=1}^{N}(z_{im} - w_{zm}t_{im})^2$$
$$- \frac{N}{2}\sum_{m=1}^{d}\log\psi_{xm} - \sum_{m=1}^{d}\frac{1}{\psi_{xm}}\sum_{i=1}^{N}(x_{im} - w_{xm}t_{im})^2$$
$$- \frac{1}{2}\sum_{m=1}^{d}\sum_{i=1}^{N}t_{im}^2 + \frac{1}{2}\sum_{m=1}^{d}\log\alpha_m - \frac{1}{2}\sum_{m=1}^{d}\alpha_m w_{zm}^2 + \frac{1}{2}\sum_{m=1}^{d}\log\alpha_m$$
$$- \frac{1}{2}\sum_{m=1}^{d}\alpha_m w_{xm}^2 + \sum_{m=1}^{d}(a_{\alpha_{m0}} - 1)\log\alpha_m - \sum_{m=1}^{d} b_{\alpha_{m0}}\alpha_m$$

The expectation of this complete data likelihood should be taken with respect to the true posterior distribution of all hidden variables $Q(\alpha, \mathbf{w_z}, \mathbf{w_x}, \mathbf{Z}, \mathbf{T})$. Unfortunately, this is an analytically intractable expression. Instead, a lower bound can be formulated by using a factorial approximation of the true posterior in terms of: $Q(\alpha, \mathbf{w_z}, \mathbf{w_x}, \mathbf{Z}, \mathbf{T}) = Q(\alpha)Q(\mathbf{w_z})Q(\mathbf{w_x})Q(\mathbf{Z}, \mathbf{T})$. While losing a small amount of accuracy, all resulting posterior distributions over hidden variables become now analytically tractable. As a final result, we now have a mechanism that infers the significance of each dimension's contribution to the observed output y and observed inputs x. We omit the resulting lengthy EM update equations for brevity and include them in the Appendix. The final regression solution regularizes over the number of retained inputs in the regression vector, performing a functionality similar to Automatic Relevance Determination (ARD) [Neal, 1994]. It is important to notice that *the resulting generalized EM updates still have a computational complexity of $O(d)$ for each EM iteration* – a level of efficiency that has not been accomplished with previous Joint Factor Analysis models, especially with one containing a full Bayesian treatment of Joint Factor Analysis.

3.3. Inference of Regression Solution

Estimating the rather complex probabilistic Bayesian model for Joint Factor Analysis reveals distributions and mean values for all hidden variables. One additional step, however, is required to infer the final regression parameters. For this purpose, we consider the predictive distribution $p(y^q|\mathbf{x^q})$ for a new noisy test input $\mathbf{x^q}$ and its unknown output y^q. We can calculate the mean of the distribution associated with $p(y^q|\mathbf{x^q})$, $\langle y^q|\mathbf{x}^q\rangle$, by conditioning y^q on $\mathbf{x^q}$ and marginalizing out all hidden variables. We can then infer the value of the regression estimate $\hat{b}$, since $\langle y^q|\mathbf{x}^q\rangle = \hat{b}^T x^q$. Since an analytical solution of the resulting integral is only possible for the probabilistic Joint Factor Analysis model in Figure 1(b) and not for the full Bayesian treatment, we restrict our computations to the simpler probabilistic model, assuming that the results will hold in approximation for the Bayesian model. Thus, we obtain:

$$p(y^q|\mathbf{x^q}, \mathbf{X}, \mathbf{Y}) = \int\int p(y^q, \mathbf{Z}, \mathbf{T}|\mathbf{x^q}, \mathbf{X}, \mathbf{Y})d\mathbf{Z}d\mathbf{T}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the training data. The resulting regression estimate, given noisy inputs $\mathbf{x^q}$ and noisy outputs y^q, is $\hat{b}_{noise}$:

$$\hat{b}_{noise} = \frac{\psi_y \mathbf{1}^T\mathbf{B}^{-1}}{\psi_y - \mathbf{1}^T\mathbf{B}^{-1}\mathbf{1}}\mathbf{\Psi}_z^{-1}\langle\mathbf{W}_z\rangle\mathbf{A}^{-1}\langle\mathbf{W}_x\rangle^T\mathbf{\Psi}_x^{-1} \quad (4)$$

where $\mathbf{\Psi}_x$ is a diagonal matrix with the vector $\psi_\mathbf{x}$ on its diagonal ($\langle\mathbf{W}_x\rangle$, $\langle\mathbf{W}_z\rangle$, $\mathbf{\Psi}_z$ are similarly defined diagonal matrices with vectors of $\langle\mathbf{w}_x\rangle$, $\langle\mathbf{w}_z\rangle$ and $\psi_\mathbf{z}$ on their diagonals, respectively), $\mathbf{A} = \left(\mathbf{I} + \left\langle\mathbf{W}_x^T\mathbf{W}_x\right\rangle\mathbf{\Psi}_x^{-1} + \left\langle\mathbf{W}_z^T\mathbf{W}_z\right\rangle\mathbf{\Psi}_z^{-1}\right)$ and $\mathbf{B} = \left(\frac{\mathbf{1}\mathbf{1}^T}{\psi_y} + \mathbf{\Psi}_z^{-1} - \mathbf{\Psi}_z^{-1}\langle\mathbf{W}_z\rangle^T\mathbf{A}^{-1}\langle\mathbf{W}_z\rangle\mathbf{\Psi}_z^{-1}\right)$. Note that Eq. (4) is similar in form to the regression estimate derived for the classical model of Joint Factor Analysis regression in Figure 1(a), which can be computed, following the same procedure, to be:

$$\hat{b}_{JFA} = \langle\mathbf{W}_z\rangle\left(\mathbf{I} + \left\langle\mathbf{W}_x^T\mathbf{W}_x\right\rangle\mathbf{\Psi}_x^{-1}\right)^{-1}\langle\mathbf{W}_x\rangle^T\mathbf{\Psi}_x^{-1} \quad (5)$$

The major difference between Eq. (4) and Eq. (5) is that the former contains an additional term $\left\langle\mathbf{W}_z^T\mathbf{W}_z\right\rangle\mathbf{\Psi}_z^{-1}$, due to the introduction of hidden variables z. The regression estimate is scaled by an additional term as well.

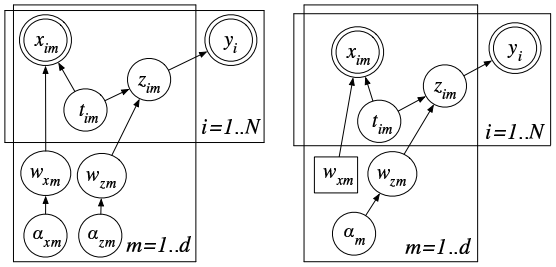
Careful observation reveals that the regression vector given by Eq. (4) is for optimal prediction from *noisy* input data. However, we are interested in obtaining the true regression vector, which is the regression vector that predicts output from *noiseless* inputs. Thus, the result in Eq. (4) is not quite suitable and what we want to calculate is the mean of $p(y^q|\mathbf{t^q})$ where $\mathbf{t^q}$ are noiseless inputs. To address this, we can take the limit of Eq. (4) by letting $\psi_\mathbf{x} \to 0$ and interpret the resulting expression to be the true regression vector for noiseless inputs (as $\psi_\mathbf{x} \to 0$, the amount of input noise approaches 0). The resulting regression vector estimate $\hat{b}_{true}$ becomes:

$$\hat{b}_{true} = \frac{\psi_y \mathbf{1}^T\mathbf{C}^{-1}}{\psi_y - \mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}\mathbf{\Psi}_z^{-1}\langle\mathbf{W}_z\rangle^T\langle\mathbf{W}_x\rangle^{-1} \quad (6)$$

where $\mathbf{C} = \left(\frac{\mathbf{1}\mathbf{1}^T}{\psi_y} + \mathbf{\Psi}_z^{-1}\right)$, which is the desired regression vector estimate for noiseless data that we use in our evaluations.

3.4. Alternative Formulations

Several alternative graphical models can be considered for the Bayesian approach of Figure 1(c). During our research, we evaluated eight different variations of the model. Two of the more interesting ones are presented here and included in our numerical comparisons below. In Figure 2(a), the precision variables over the regression parameters w_{zm} and w_{xm} are decoupled: instead of sharing a common α_m, there is a precision variable α_{zm} over w_{zm} and a precision variable α_{xm} over w_{xm}. This model does not enforce the interdependency between w_{zm} and w_{xm} as stringently as our model in Figure 1(c). Figure 2(b) shows yet another variation of the model, with a precision variable α_m over the regression parameter w_{zm} only. In this formulation, w_{zm} is a point-estimated parameter. This model ignores the effect of $\mathbf{w_x}$ and only regularizes the $\mathbf{w_z}$ branch.



(a) Formulation I (b) Formulation II

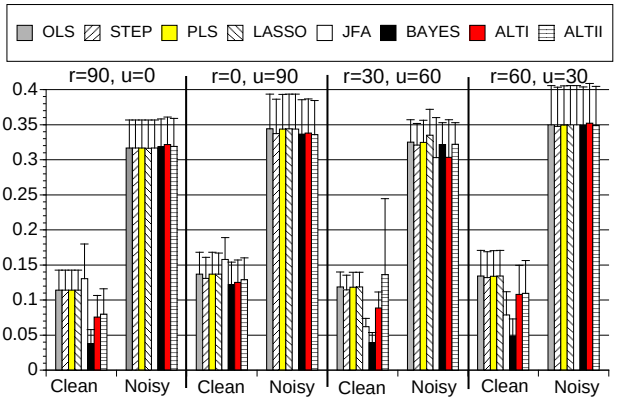
Figure 2. Graphical Models for Alternative Formulations of the Bayesian de-noising version of JFA. Random variables are in circular nodes, observed random variables are in double circles, and point estimated parameters are in square nodes. d is the total number of input dimensions while N is the total number of samples in the data set.

4. Evaluation

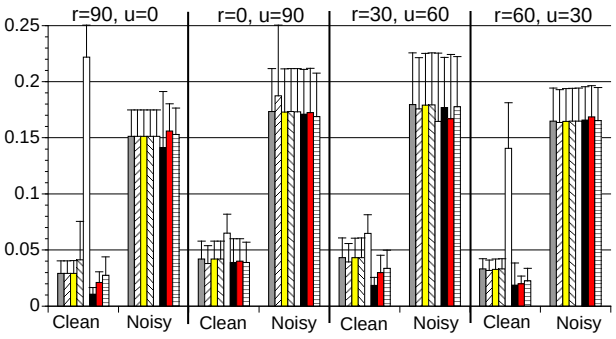
We applied our algorithm on both synthetic data and robotic data for the problem of accurate prediction. The goal of these evaluations was to determine how much better our Bayesian de-noising algorithm fared compared to other standard techniques in terms of generalization performance on high dimensional, ill-conditioned noisy data. First we evaluate our algorithm on a synthetic data set. Then, to illustrate the algorithm on a real-world application, we apply it to a robotic vision head for the task of estimation of the rigid body dynamics model parameters.

4.1. Synthetic Data Set

We synthesized random input training data consisting of 10 relevant dimensions and 90 irrelevant and redundant dimensions. The first 10 input dimensions were drawn from a multi-dimensional Gaussian distribution with a random covariance matrix. The output data was generated from the relevant input data using the ordered vector $b_{true} = [1, 2, ..., 10]^T$. A signal-to-noise ratio (SNR) of 5 was then added to the outputs. Next, the input data was made noisy by adding Gaussian noise with varying SNRs (a SNR of 2 for strongly noisy input data and a SNR of 5 for less noisy input data) to the relevant 10 input dimensions. A varying number of redundant data vectors was added to the input data, generated from random convex combinations of the 10 noisy relevant data vectors. Finally, we added irrelevant data columns until a total of 100 input dimensions were reached, creating training input data that contained irrelevant and redundant dimensions. The irrelevant data was drawn from a Normal$(0, 1)$ distribution.



(a) $\text{SNR}_x = 2$, $\text{SNR}_y = 5$



(b) $\text{SNR}_x = 5$, $\text{SNR}_y = 5$

Figure 3. Normalized mean squared errors on noiseless (clean) test data and noisy test data for a 100 dimensional data set with 10 relevant input dimensions and various combinations of redundant dimensions r and irrelevant input dimensions u, averaged over 10 trials, for different levels of noisy data.

The test data set was created in a similar manner except that the input data and output data were left noise-free. A second test data set consisting of noisy input and output data, possessing the same noise characteristics as the training data set, was also generated.

We compared our Bayesian de-noising algorithm with the following methods: OLS regression; stepwise re-

gression [Draper & Smith, 1981], which tends to be inconsistent in the presence of collinear inputs [Derksen & Keselman, 1992]); Partial Least Squares regression (PLS) [Wold, 1975], a slightly heuristic but empirically quite successful regression method for high dimensional data; LASSO regression [Tibshirani, 1996], which gives sparse solutions by shrinking certain regression coefficients to 0 under the control of a manually set tuning parameter; our probabilistic treatment of Joint Factor Analysis in Figure 1(b); and the two alternative Bayesian formulations described in Section 3.4. Figure 3 shows the generalization performances of all the algorithms for data sets with varying redundant dimensions r and irrelevant dimensions u, averaged over 10 runs per condition. Our Bayesian de-noising algorithm performs best for predictions on noiseless data, with 10 to 70% improvement in generalization performance for data containing an input SNR of 2 (i.e. strongly noisy data) and a smaller but still significant improvement of 7 to 50% for less noisy data (SNR of 5), as the small black bars show. We can see that Joint Factor Analysis, due to the lack of regularization, degrades in the presence of many redundant dimensions, r – an effect that is emphasized on less noisy input data (i.e. input SNR of 5 on data with $r = 90, u = 0$ and $r = 60, u = 30$). The two alternative formulations of the Bayesian model also fail to offer improved performances over input data of various noise levels. Notice that all algorithms perform equally badly when predicting on noisy test data. Thus, the Bayesian de-noising algorithm is only advantageous for applications where predictions on noiseless data are desired.

4.2. Robotic Oculomotor Vision Head

Next, we move on to a sample application: a 7 DOF robotic vision head manufactured by Sarcos as shown in Figure 4, possessing 3 DOFs in the neck and 2 DOFs for each eye. With 11 features per DOF, this gives a total of 77 features. The kinematic structure of robotic systems always creates non-identifiable parameters and thus, redundancies [An et al., 1988]. For the robotic vision head, there are 9 non-identifiable parameters if the training set is full rank. Due to the nature of data collection, the training set can be less than full rank and often is, giving a high-dimensional data set that contains redundancy. The robot is controlled at 420 Hz with a vxWorks real-time operating system running out of a VME bus. We collected about 500,000 data points from the robotic system while it performed sinusoidal movements with varying frequencies and phase offsets in all DOFs.

Figure 4. Sarcos Robotic Oculomotor Vision Head

The problem at hand involves parameter estimation of the rigid body dynamics (RBD) model of the robotic system, which consists of 11 parameters for each DOF: one mass parameter, three center of mass parameters, six inertial parameters (the upper triangular matrix of the symmetric inertia matrix, relative to the center of mass) (cf. [An et al., 1988]), and one viscous friction parameter. If the data contains no redundancy, in theory, there is only one true solution for the RBD parameters and no ambiguity exists. However, this RBD system identification task is not so straightforward to resolve due to: i) noise in the input data, ii) insufficiently rich data to allow identifiability of all RBD parameters (i.e., the data is ill-conditioned), and iii) the need for physical consistency constraints on the RBD parameters. The constraints on the RBD parameters are given by the positive definite inertia matrices and the parallel axis theorem and these constraints are highly nonlinear.

To enforce the nonlinear constraints in a linear way, we assume that the parameter vector θ is generated from virtual parameters $\hat{\theta}$, as given for one DOF below:

$$
\begin{aligned}
&\theta_1 = \hat{\theta}_1^2, \theta_2 = \hat{\theta}_2\hat{\theta}_1^2, \theta_3 = \hat{\theta}_3\hat{\theta}_1^2, \theta_4 = \hat{\theta}_4\hat{\theta}_1^2, \theta_{11} = \hat{\theta}_{11}^2 \\
&\theta_5 = \hat{\theta}_5^2 + \left(\hat{\theta}_4^2 + \hat{\theta}_3^2\right)\hat{\theta}_1^2 \\
&\theta_6 = \hat{\theta}_5\hat{\theta}_6 - \hat{\theta}_2\hat{\theta}_3\hat{\theta}_1^2, \theta_7 = \hat{\theta}_5\hat{\theta}_7 - \hat{\theta}_2\hat{\theta}_4\hat{\theta}_1^2 \\
&\theta_8 = \hat{\theta}_6^2 + \hat{\theta}_8^2 + \left(\hat{\theta}_2^2 + \hat{\theta}_4^2\right)\hat{\theta}_1^2 \\
&\theta_9 = \hat{\theta}_6\hat{\theta}_7 + \hat{\theta}_8\hat{\theta}_9 - \hat{\theta}_3\hat{\theta}_4\hat{\theta}_1^2 \\
&\theta_{10} = \hat{\theta}_7^2 + \hat{\theta}_9^2 + \hat{\theta}_{10}^2 + \left(\hat{\theta}_2^2 + \hat{\theta}_3^2\right)\hat{\theta}_1^2
\end{aligned} \tag{7}
$$

In essence, these virtual parameters $\hat{\theta}$ correspond to the square root of the mass, the true center-of-mass coordinates (i.e., not multiplied by the mass), the six inertial parameters describing the inertia matrix at the DOF's center of gravity, and the square root of the viscous friction coefficient. The functions in Eq. (7) encode the parallel axis theorem and some additional constraints, essentially ensuring that mass and viscous friction coefficients remain strictly positive. Given the above formulation, any arbitrary set of virtual parameters gives rise to a physically consistent set of actual parameters for the RBD problem. For a robotic system with s DOFs, Eq. (7) is repeated for each DOF. The result is a regression vector θ with $d = 11s$ dimensions. All correlations between DOFs are taken into account by means of complex basis function expansion.

Our Bayesian de-noising algorithm (as well as any

Table 1. Root mean squared errors for position (in radians), velocity (radians/sec) and feedback command (in Newton-meters) for ridge regression with nonlinear gradient descent, our Bayesian de-noising algorithm, LASSO regression with the projection step, and stepwise regression with the projection step. Standard deviations were negligible and thus omitted.

Algorithm	Position (rad)	Velocity (rad/s)	Feedback (Nm)
Ridge Regression	0.0291	0.2465	0.3969
Bayesian De-noising	0.0243	0.2189	0.3292
LASSO Regression	0.0308	0.2517	0.4274
Stepwise Regression	FAILURE	FAILURE	FAILURE

other traditional RBD parameter estimation method) generates the parameter vector θ, not the virtual parameters $\hat{\theta}$. Hence, to ensure that our final parameters satisfy the constraints of Eq. (7), we added a post-processing step that projects the result of the Bayesian algorithm onto these constraints. Again, as in the Bayesian EM algorithm, this post-processing step attempts to maximize the lower bound on the expected complete log likelihood, an optimization that is performed by gradient descent with respect to the virtual parameters $\hat{\theta}$. However, this is a very large optimization problem and doing this for each EM iteration is computationally complex and burdensome. Instead, we take the regularized robust solution that the Bayesian algorithm produces and find the optimal point-estimates of the regression vector. We do this by holding the all variables constant except for $\mathbf{w_z}$ and performing an M-step via a Maximum Likelihood approach. We use Eq. (6) to express $\mathbf{w_z}$ in terms of $\hat{b}_{true}$ (which also happens to be θ, the regression parameters). Then, we simply substitute this expression for $\mathbf{w_z}$ into the lower bound of the expected complete log likelihood. Finally, we perform gradient descent on the virtual parameters $\hat{\theta}$ until they converge. The resulting regression vector estimate θ (i.e., $\hat{b}_{true}$) produces physically consistent RBD parameters.

We compared our Bayesian algorithm with 3 other techniques for parameter estimation on the robot data. The first technique consisted of ridge regression using a hand-tuned regularization parameter with nonlinear gradient descent performed on the virtual parameters of the system. The second algorithm was a version of LASSO regression that had the additional step of projecting the resulting parameter values onto the constraint space to produce physically consistent RBD parameters. Finally, the last algorithm was a version of stepwise regression with the additional projection step. All four algorithms produced physically consistent RBD parameters. Note that the other algorithms used in the synthetic data set like PLS and JFA were not applied, since they fail to explicitly eliminate irrelevant input features and do not perform any form of reasonable parameter identification.

For evaluation, we implemented a computed torque control law on the robot, using the estimated parameters from each technique. Results are quantified as the root mean squared errors in position tracking, velocity tracking and the root mean squared feedback command. Table 1 shows these results averaged over all 7 DOFs. The standard deviations on these results were omitted since they were so small, due to the repeatability of the robotic system. The Bayesian parameter estimation approach performed around 10 to 20% better than the ridge regression with gradient descent approach, thus validating the effectiveness of our methods. LASSO regression performed worse than ridge regression with gradient descent. Unsurprisingly, stepwise regression produced RBD parameters that were so physically off that they were impossible to run on the robotic head. This can be explained by stepwise regression's failure to identify the relevant features in the data set, resulting in RBD parameters that were just wrong.

5. Conclusion

We derived a Bayesian linear regression algorithm that is robust to high dimensional ill-conditioned data contaminated with noisy inputs and noisy outputs. The Bayesian de-noising algorithm outperforms alternative methods on synthetic data, with a 10 to 70% improvement in generalization on noiseless data. As a sample application, we demonstrated the efficiency of the algorithm by applying it on a 7 DOF robotic head for the task of system identification. Our algorithm successfully identified the system parameters with 10 to 20% higher accuracy than other standard techniques. Our suggested technique can serve as a drop-in replacement for many linear regression methods and can be inserted into nonlinear regression models that have a linear parameterization such as locally weighted regression, mixture of experts or radial basis function networks.

Acknowledgments

This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, a NASA grant AC#98 − 516, an AFOSR grant on Intelligent Control, the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

References

An, C. H., Atkeson, C. G., & Hollerbach, J. M. (1988). *Model-based control of a robot manipulator*. MIT Press.

Atkeson, C. G., Moore, A., & Schaal, S. (1997). Locally weighted learning. In *Artificial intelligence review*, vol. 11, 11–73. Kluwer.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society. Series B*, *39*, 1–38.

Derksen, S., & Keselman, H. (1992). Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, *45*, 265–282.

Draper, N. R., & Smith, H. (1981). *Applied regression analysis*. Wiley.

D'Souza, A., Vijayakumar, S., & Schaal, S. (2004). The bayesian backfitting relevance vector machine. In *Proceedings of the 21st international conference on machine learning*. ACM Press.

Ghahramani, Z., & Beal, M. (2000). Graphical models and variational methods. In D. Saad and M. Opper (Eds.), *Advanced mean field methods - theory and practice*. MIT Press.

Golub, G. H., & Van Loan, C. (1989). *Matrix computations*. John Hopkins University Press.

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models*. No. 43 in Monographs on Statistics and Applied Probability. Chapman and Hall.

Hollerbach, J. M., & Wampler, C. W. (1996). The calibration index and the role of input noise in robot calibration. In G. Giralt and G. Hirzinger (Eds.), *Robotics research: The seventh international symposium*, 558–568. Springer.

Massey, W. (1965). Principal component regression in exploratory statistical research. *Journal of the American Statistical Association*, *60*, 234–246.

Neal, R. (1994). *Bayesian learning for neural networks*. Doctoral dissertation, Dept. of Computer Science, University of Toronto.

Rao, Y. N., & Principe, J. (2002). Efficient total least squares method for system modeling using minor component analysis. In *Proceedings of international workshop on neural networks for signal processing*, 259–268. IEEE.

Strassen, V. (1969). Gaussian elimination is not optimal. *Num Mathematik*, *13*, 354–356.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society, Series B*, *58*, 267–288.

Van Huffel, S., & Vanderwalle, J. (1991). The total least squares problem: Computational aspects and analysis. *Society for Industrial and Applied Mathematics*.

Wold, H. (1975). Soft modeling by latent variables: The nonlinear iterative partial least squares approach. In J. Gani (Ed.), *Perspectives in probability and statistics, papers in honor of m.s. bartlett*. Academic Press.

A. Appendix

We can then derive the following EM updates using standard manipulations of normal distributions:

M-step :

$$\psi_y = \frac{1}{N}\sum_{i=1}^{N}\left(y_i^2 - 2\mathbf{1}y_i\left\langle z_i\right\rangle + \mathbf{1}^T\left\langle z_i z_i^T\right\rangle\mathbf{1}\right)$$

$$\psi_{zm} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\langle z_{im}^2\right\rangle - 2\left\langle w_{zm}\right\rangle\left\langle z_{im}t_{im}\right\rangle + \left\langle w_{zm}^2\right\rangle\left\langle t_{im}^2\right\rangle\right)$$

$$\psi_{xm} = \frac{1}{N}\sum_{i=1}^{N}\left(x_{im}^2 - 2\left\langle w_{xm}\right\rangle\left\langle t_{im}\right\rangle x_{im} + \left\langle w_{xm}^2\right\rangle\left\langle t_{im}^2\right\rangle\right)$$

E-step :

$$\sigma_{w_{zm}}^2 = \frac{1}{\frac{1}{\psi_{zm}}\sum_{i=1}^{N}\left\langle t_{im}^2\right\rangle + \left\langle\alpha_m\right\rangle}, \left\langle w_{zm}\right\rangle = \frac{\sigma_{w_{zm}}^2}{\psi_{zm}}\sum_{i=1}^{N}\left\langle z_{im}t_{im}\right\rangle$$

$$\sigma_{w_{xm}}^2 = \frac{1}{\frac{1}{\psi_{xm}}\sum_{i=1}^{N}\left\langle t_{im}^2\right\rangle + \left\langle\alpha_m\right\rangle}, \left\langle w_{xm}\right\rangle = \frac{\sigma_{w_{xm}}^2}{\psi_{xm}}\sum_{i=1}^{N}x_{im}\left\langle t_{im}\right\rangle$$

$$\hat{a}_{\alpha_m} = a_{\alpha_{m0}} + 1, \hat{b}_{\alpha_m} = b_{\alpha_{m0}} + \frac{\left\langle w_{zm}^2\right\rangle + \left\langle w_{xm}^2\right\rangle}{2}$$

The covariance matrix, Σ, of the joint posterior distribution of $\mathbf{Z}$ and $\mathbf{T}$ is $\begin{bmatrix}\Sigma_{zz} & \Sigma_{zt}\\ \Sigma_{tz} & \Sigma_{tt}\end{bmatrix}$, where:

$$\Sigma_{zz} = \mathbf{M} - \frac{\mathbf{M}\mathbf{1}\mathbf{1}^T\mathbf{M}}{\psi_y + \mathbf{1}^T\mathbf{M}\mathbf{1}}, \Sigma_{zt} = -\Sigma_{zz}\left\langle\mathbf{W}_z\right\rangle\mathbf{\Psi}_z^{-1}\mathbf{K}^{-1}, \Sigma_{tz} = \Sigma_{zt}^T$$

$$\Sigma_{tt} = \mathbf{K}^{-1} + \mathbf{K}^{-1}\left\langle\mathbf{W}_z\right\rangle^T\mathbf{\Psi}_z^{-1}\Sigma_{zz}\mathbf{\Psi}_z^{-1}\left\langle\mathbf{W}_z\right\rangle\mathbf{K}^{-1}$$

$$\mathbf{K} = \mathbf{I} + \left\langle\mathbf{W}_x^T\mathbf{W}_x\right\rangle\mathbf{\Psi}_x^{-1} + \left\langle\mathbf{W}_z^T\mathbf{W}_z\right\rangle\mathbf{\Psi}_z^{-1}$$

$$\mathbf{M} = \mathbf{\Psi}_z + \left\langle\mathbf{W}_z\right\rangle\left(\mathbf{I} + \left\langle\mathbf{W}_x^T\mathbf{W}_x\right\rangle\mathbf{\Psi}_x^{-1} + \left(\Sigma_{\mathbf{W}_z}\right)_{mm}\mathbf{\Psi}_z^{-1}\right)^{-1}\left\langle\mathbf{W}_z\right\rangle^T$$

and where $\left\langle\mathbf{W}_x\right\rangle$ is a diagonal d by d matrix with $\left\langle\mathbf{w_x}\right\rangle$ along its diagonal. Similarly, $\left\langle\mathbf{W}_z\right\rangle$, $\mathbf{\Psi}_x$, $\mathbf{\Psi}_z$ are d by d diagonal matrices with diagonal vectors of $\left\langle\mathbf{w_z}\right\rangle, \psi_\mathbf{x}$ and $\psi_\mathbf{z}$. The E-step updates for $\mathbf{Z}$ and $\mathbf{T}$ are then:

$$\left\langle z_i\right\rangle = \frac{y_i}{\psi_y}\mathbf{1}^T\Sigma_{zz} + x_i\left\langle\mathbf{W}_x\right\rangle^T\mathbf{\Psi}_x^{-1}\Sigma_{tz}$$

$$\left\langle t_i\right\rangle = \frac{y_i}{\psi_y}\mathbf{1}^T\Sigma_{zz}\left\langle\mathbf{W}_z\right\rangle\mathbf{\Psi}_z^{-1}\mathbf{K}^{-1} + x_i\left\langle\mathbf{W}_x\right\rangle^T\mathbf{\Psi}_x^{-1}\Sigma_{tt}$$

$$\sigma_z^2 = \mathrm{diag}(\Sigma_{zz}), \sigma_t^2 = \mathrm{diag}(\Sigma_{tt}), \mathrm{cov}(z,t) = \mathrm{diag}(\Sigma_{zt})$$