Statistical Solutions For and From Signal Processing

by

Luke Bornn

B.Sc., University of the Fraser Valley, 2003

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

The Faculty of Graduate Studies

(Statistics)

The University Of British Columbia

(Vancouver)

August, 2008

Abstract

With the wide range of fields engaging in signal processing research, many methods do not receive adequate dissemination across disciplines due to differences in jargon, notation, and level of rigor. In this thesis, I attempt to bridge this gap by applying two statistical techniques originating in signal processing to fields for which they were not originally intended. Firstly, I employ particle filters, a tool used for state estimation in the physics signal processing world, for the task of prior sensitivity analysis and cross-validation in Bayesian statistics. Secondly, I demonstrate the application of support vector forecasters, a tool used for forecasting in the machine learning signal processing world, to the field of structural health monitoring.

Table of Contents

List of Figures

Acknowledgements

I am indebted to my mentors, Dr. Arnaud Doucet and Dr. Raphael Gottardo, for the wealth of opportunities and immense support they have provided me. I am also grateful to the many people I have had a chance to interact with at both the University of British Columbia and Los Alamos National Labs. In particular I thank Dr. Dave Higdon and Dr. Todd Graves for guiding me down a fruitful path.

Statement of Co-Authorship

Excluding chapters 2 and 3, all material in this thesis was solely authored.

While I identified and carried out the research in chapter 2, Arnaud Doucet and Raphael Gottardo provided much guidance, criticism, and feedback.

The data in chapter 3 was produced by Gyuhae Park and Kevin Farinholt. Chuck Farrar assisted in writing some of the details pertaining to the structural health monitoring field. In addition, his guidance and feedback contributed immensely to the development of the work.

Chapter 1

Introduction

The research area of signal processing is concerned with analyzing signals including sound, video, and radar. There are many components to this task, including storage and compression, removing noise, and extracting features of interest. As an example, we might have a noisy recording of a telephone conversation for which we want to store the signal, remove the noise, and identify the speakers. These signals can take many forms, either digital or analog. We focus on statistical signal processing, which is concerned with studying signals based on their statistical properties. We begin by describing two statistical methods employed for signal processing, the first being particle filtering, and the latter being support vector forecasters. Because chapter 3 contains a detailed development of support vector forecasters, we forego these details here. Later chapters then extend these methodologies to fields for which they were not intended, specifically prior sensitivity analysis and cross validation as well as structural health monitoring

1.1 Particle Filtering

One of the crucial areas of study in signal processing is filtering, which is concerned with estimating a dynamic system's true state from a series of noisy measurements. Specifically, we assume that the system dynamics are known up to some parameter(s). The underlying state-space model may be written as

$$x_t|x_{t-1} \sim p_{x,t}(x|x_{t-1})$$
$$y_t|x_t \sim p_{y,t}(y|x_t)$$

where x_t and y_t denote the unobserved state and observation at time t, respectively; $p_{x,t}$ and $p_{y,t}$ are the state transition and measurement models, respectively. Also, we assume a prior distribution $p(x_0)$ on x_0. In the case of linearly additive noise, we may write this state-space model as

$$x_t = f(x_{t-1}|\theta) + \eta_t$$
$$y_t = h(x_t) + \epsilon_t. \tag{1.1}$$

Here both the stochastic noise η_t and the measurement noise ϵ_t are mutually independent and identically distributed sequences with known density functions. In addition, $f(x_{t-1}|\theta)$ and $h(x_t)$ are known functions up to some parameters θ.

In order to build the framework on which to describe the filtering methodologies, we first frame the above state-space model as a recursive Bayesian estimation problem. Specifically, we are interested in obtaining the posterior distribution

$$p(x_{0:t}|y_{1:t}) \tag{1.2}$$

where $x_{0:t} = \{x_0, x_1, \ldots, x_t\}$ and $y_{1:t} = \{y_1, y_2, \ldots, y_t\}$. Often we don't require the entire posterior distribution, but merely one of its marginals. For instance, we are often interested in the estimate of state given all observations up to that point; we call this distribution the filtering density and denote it as

$$p(x_t|y_{1:t}). \tag{1.3}$$

By knowing this density, we are able to make estimates about the system's state, including measures of uncertainty such as confidence intervals.

If the functions f and h are linear and both η_t and ϵ_t are Gaussian, Kalman filtering is able to obtain the filtering distribution in analytic form. In fact it can be seen that all of the distributions of interest are Gaussian with means and covariances that can be simply calculated. However, when the dynamics are non-linear or the noise non-Gaussian, alternative methods must be used.

In the case of non-linear dynamics with Gaussian noise, the standard methodology is the extended Kalman filter, which may be considered as a nonlinear Kalman filter which linearizes around the current mean and covariance. However, as a result of this linearization, the filter may diverge if the initial state estimate is wrong or the process is incorrectly modeled. In addition, the calculation of the Jacobian in the extended Kalman filter can become tedious in high-dimension problems. One attempted solution to this problem has been the unscented Kalman filter (Wan and van der Merwe, 2001), which approximates the nonlinearity by transforming a random variable instead of through a Taylor expansion, as the extended Kalman filter does. By employing a deterministic sampling technique known as the unscented transform (Julier and Uhlmann, 1997), UKF selects a minimal set of sample points around the mean which are then propagated through the non-linear functions while recovering the covariance matrix.

When either the stochastic or measurement noise is non-Gaussian, Monte Carlo methods must be employed, in particular particle filters. This Monte

Carlo based filtering method relies on a large set of samples, called particles, which are evolved through the system dynamics with potentially non-Gaussian noise using importance sampling and bootstrap techniques. At each time step the empirical distribution of these particles is used to approximate the distribution of interest and its associated features. By sampling from some proposal distribution $q(x_{0:t}|y_{1:t})$ in order to approximate (1.2), we may use importance sampling with corresponding unnormalized weights

$$w_t = \frac{P(y_{1:t}|x_{0:t})P(x_{0:t})}{q(x_{0:t}|y_{1:t})}.$$

However, we typically wish to perform this estimation sequentially, and hence we can take advantage of the Markov nature of the state and measurement process along with proposal distributions of the form $q(x_{0:t}|y_{1:t}) = q(x_{0:t-1}|y_{1:t-1})q(x_t|x_{0:t-1}, y_{1:t})$. From this we obtain the recursive weight formula

$$w_t = w_{t-1}\frac{P(y_t|x_t)P(x_t|x_{t-1})}{q(x_t|x_{0:t-1}, y_{1:t})}.$$

This equation allows for the sequential updating of importance weights given an appropriate choice of proposal distribution $q(x_t|x_{0:t-1}, y_{1:t})$, as well as simple calculation of the filtering density (1.3). Since we can sample from this proposal distribution and evaluate the likelihood and transition probabilities, the particle filter simply involves generating a prior set of samples, evolving these samples forward with the proposal distribution, and subsequently calculating the importance weights. In addition, to prevent particle degeneracy, we employ a resampling step to remove particles with low weight and multiply those with high weight (Douc et al., 2005).

The choice of proposal distribution has a significant effect on the rate of degeneracy. The standard (and simplest) choice is the prior distribution $q(x_t|x_{0:t-1}, y_{1:t}) = P(x_t|x_{t-1})$ since the weights simplify to a calculation of the likelihood. However, if the likelihood is not near the prior, this choice will lead to large variance in the importance weights, and hence we would like to employ a proposal distribution which uses the data to provide a better estimate of the posterior distribution. One such possibility is to use a Gaussian approximation of the posterior as the proposal distribution (van der Merwe et al., 2001).

Often the problem of filtering isn't restricted to the estimation of state, but is also concerned with estimating some parameters θ of the dynamic model $f(x|\theta)$. Further complicating matters, the only information we have about the state and the model parameters is the noisy measurements $\{y_t\}_{t\geq 1}$. While there are several approaches for solving this problem, we focus on dual

estimation, namely the use of parallel filters to estimate state and model parameters (Wan et al., 2000). Specifically, a state-space representation is used for both the state and parameter estimate problems. While the state-space representation for the state is given in equation (1.1), the representation of the model parameters is given by

$$\begin{aligned}\theta_t &= \theta_{t-1} + \nu_t \\ y_t &= f(x_{t-1}|\theta_t) + \eta_t + \epsilon_t.\end{aligned}$$

Here η_t and ϵ_t are as in (1.1), while ν_t is an additional iid noise term. Thus we can run two parallel filters for both state and parameters. At each time step the current state estimate is used in the parameter filter and the current parameter estimate is used in the state filter. The situation is complicated in the particle filter situation, due to the well-known problem of degenerate weights (Casarin and Marin, 2008).

Through this filtering methodology we are able to estimate the state of a dynamic system from noisy measurements, as well as the associated uncertainty of these estimates. In addition, the dual framework provides a mechanism for estimating model parameters along with the state. These filtering tools approximate a sequence of distributions of increasing dimension. In later chapters, we show how the particle filtering methodology may be adapted for situations involving distributions of equal dimension, and subsequently build an algorithm for efficiently performing prior sensitivity analysis and cross-validation.

1.2 Support Vector Forecasters

While particle filtering and other filtering methods rely on knowledge of the underlying process to de-noise the signal, support vector regression forecasters have a slightly different purpose. Specifically, they use a training data set to build a model of the signal, which is then used to predict subsequent pieces of the signal. The previous section contains a thorough description of filtering since the associated manuscript of chapter 2 bypasses this development. However, chapter 3 provides a thorough development of support vector forecasters, and hence we forego this development here, instead simply providing some useful references. The recent work of Steinwart and Christmann (2008) provides thorough details on both theoretical and applied aspects of support vector machines, while Schlkopf and Smola (2001) contains details on kernel-based learning.

1.3 References

Casarin, R., Marin, J-M. (2008). "Online data processing: comparison of Bayesian regularized particle filters." arXiv:0806.4242v1.

Douc, R., Cappe, O., Moulines, E. (2005). "Comparison of resampling schemes for particle filtering." *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis.* pp 6469.

Julier, S. and Uhlmann, J. (2007). "A new extension of the kalman lter to nonlinear systems." *Proceedings of AeroSense: The 11th Symposium on Aerospace/Defence Sensing, Simulation and Controls.*

Scholkopf, B. and Smola, A.J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines.* Springer, New York.

van der Merwe, R., Doucet, A., de Freitas, N., Wan, E. (2001). "The unscented particle lter." *Advances in Neural Information Processing Systems.* 13:584590.

Wan, E. and van der Merwe, R. (2001). "The unscented kalman lter." *Kalman Filtering and Neural Networks.* Ch. 7. Ed. S. Haykin.

Wan, E., van der Merwe, R., Nelson, A. (2000). "Dual estimation and the unscented transformation." *Advances in Neural Information Processing Systems.* 12:666672.

Chapter 2

An Efficient Computational Approach for Prior Sensitivity Analysis and Cross-Validation

2.1 Introduction and Motivation

[1]An important step in any Bayesian analysis is to assess the prior distribution's influence on the final inference. In order to check prior sensitivity, the posterior distribution must be studied using a variety of prior distributions. If these posteriors are not available analytically, they are usually approximated using Markov chain Monte Carlo (MCMC). Since obtaining the posterior distribution for one given prior can be very expensive computationally, repeating the process for a large range of prior distributions is often prohibitive. Importance sampling has been implemented as an attempted solution (Besag et al., 1995), but the potential of infinite variance importance weights makes this technique useless if the posterior distribution changes more than a trivial amount as the prior is altered. Additionally, this importance weight degeneracy increases with the dimension of the parameter space.

One such prior sensitivity problem is the creation of regularization path plots – a commonly used tool when performing penalized regression. In these situations there is typically a tuning parameter which controls the amount of shrinkage on the regression coefficients; regularization path plots graphically display this shrinkage as a function of the tuning parameter. For instance, in the LASSO shrinkage and variable selection method of Tibshirani (1996), the LARS algorithm (Efron et al., 2004) may be employed to quickly produce

[1]A version of this chapter has been submitted for publication. Bornn, L., Doucet, A., Gottardo, R. "An Efficient Computational Approach for Prior Sensitivity Analysis and Cross-Validation."

these plots. In the Bayesian version (Vidakovic, 1998; Park and Casella, 2008), however, we may want to plot the posterior means of the regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ for a range of the tuning (or penalty) parameter λ. The corresponding posterior distributions are proportional to

$$\exp\left(-\frac{1}{2\sigma^2}\left[(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - \lambda\sum_{j=1}^{p}|\beta_j|\right]\right) \tag{2.1}$$

where the response $\mathbf{y}$ is assumed to come from a normal distribution with mean $\mathbf{X}\boldsymbol{\beta}$ and variance σ^2 for a model matrix $\mathbf{X}$. Since approximating (2.1) using MCMC at one level of λ can take upwards of an hour depending on the precision required, producing this plot by repeating MCMC hundreds of times for different λ is impractical.

Another tool requiring repeated posterior approximations is cross-validation, which has two primary statistical purposes. The first is finding the value of a given parameter (for instance, the penalty parameter in penalized regression) which minimizes prediction error. The second is comparing different models' or methodologies' prediction performance. In both situations the data is split into a training set, which is used to fit the model, and a testing set, which is used to gauge the prediction performance of the trained model. A typical example would involve fitting a model on the training set for a range of values of some model parameter, then setting this parameter to the value that results in the lowest prediction error rate on the testing set. For example, we might wish to select the value of λ in (2.1) to minimize prediction error. From a computational standpoint, cross-validation is similar to prior sensitivity in both structure and complexity. Further, the entire process is usually repeated for a variety of different training and testing sets and the results are then combined. Although importance sampling has been applied to cross-validation (for example, Alqallaf and Gustafson, 2001), the problem of infinite variance importance weights remains (Peruggia, 1997).

In this paper, we begin by motivating and developing sequential Monte Carlo (SMC) methods, then subsequently apply them to prior sensitivity analysis and cross-validation. In Section 2 we develop an efficient algorithm for sampling from a sequence of potentially quite similar probability distributions defined on a common space. Section 3 demonstrates the algorithm in a prior sensitivity setting and applies it to the creation of regularization path plots and the sensitivity of the tuning parameters when performing variable selection using g-Priors. Cross-validation with application to Bayesian penalized regression is developed in Section 4. We close with extensions and concluding remarks in Section 5.

2.2 Sequential Monte Carlo Algorithms

SMC methods are often used in the analysis of dynamic systems where we are interested in approximating a sequence of probability distributions $\pi_t(\theta_t)$ where $t = 1, 2, 3, .., T$. The variable θ_t can be of evolving or static dimension as t changes; note that t is simply an index variable and need not be real time. Most work in the SMC literature is interested in the evolving dimension case, with applications to state-space models (Doucet et al., 2000) and target tracking (Liu and Chen, 1998) among others. The static case, where each π_t lies in a common space, has received less attention (Chopin, 2002; Del Moral et al., 2006). The goal of SMC methods is to sample from the distributions $\{\pi_t\}$ sequentially, i.e. first from π_1, then π_2, up to π_T. In some situations we are concerned with each intermediate distribution, whereas in others only the final distribution π_T is of interest (for example, Neal, 2001). For further reading, the edited volume of Doucet et al. (2001) covers a range of developments in SMC theory and application.

The situation where the sequence of distributions lie in a common space arises in several applications. For instance, the number of observations in some experiments can make MCMC prohibitive. In this case π_t might be the posterior distribution of a parameter given the observations 1 through t. Moving through the data with a sequential strategy in this way may decrease computational complexity. Another application is transitioning from a simple distribution π_1 to a more complex distribution of interest π_T. Alternatively we could consider situations analogous to simulated annealing (Kirkpatrick et al., 1983), where $\pi_t(\theta) \propto [\pi(\theta)]^{\phi_t}$ for an increasing sequence $\{\phi_t\}$, $t = 1, 2, 3, ..., T$. In all of these examples the bridging distributions $\pi_1, \ldots, \pi_{T-1}$ are only used to reach the final distribution of interest π_T. When we are interested in a certain feature of each π_t, SMC will typically be computationally cheaper than MCMC even if we can successfully sample from each π_t using MCMC. This is because SMC borrows information from adjacent distributions, using the samples from earlier distributions to help in approximating later distributions. Often the difficulty in using SMC is constructing this sequence of distributions; both prior sensitivity and cross-validation are situations where there exists a natural sequence upon which SMC may be applied. From here forward we assume the distributions to have a common support.

For all times t, we seek to obtain a collection of N weighted samples (called particles) $\{W_t^{(i)}, \theta_t^{(i)}\}$, $i = 1, ..., N$ approximating π_t where the weights are positive and normalized to sum to 1. We may estimate expected values with these particles using $\hat{E}_{\pi_t}(g(\theta)) = \sum_{i=1}^{N} W_t^{(i)} \cdot g(\theta_t^{(i)})$. One tech-

nique used in SMC is importance sampling, where particles $\{W_{t-1}^{(i)}, \theta_{t-1}^{(i)}\}$ distributed as π_{t-1} may be reused, reweighting them (before normalization) according to

$$W_t^{(i)} \propto W_{t-1}^{(i)} \times \frac{\pi_t(\theta_{t-1}^{(i)})}{\pi_{t-1}(\theta_{t-1}^{(i)})} \tag{2.2}$$

in order to obtain an approximation of π_t. Thus we obtain the current weights by multiplying the previous weights by an incremental weight.

In an attempt to prevent these weights from becoming overly non-uniform, we may move each particle $\theta_{t-1}^{(i)}$ (currently distributed according to π_{t-1}) with a Markov kernel $K_t(\theta, \theta')$ to a new position $\theta^{(i)'}$, then subsequently reweight the moved particles to be distributed according to π_t. Although the kernel $K_t(\theta, \theta') = \pi_t(\theta')$ minimizes the variance of the importance weights, it is typically impossible to sample from; thus it has been proposed to use Markov kernels with invariant distribution π_t (Gilks and Berzuini, 2001). A direct application of this strategy suffers from a major flaw, however, as the importance distribution given by

$$\eta_t(\theta_t) = \int \pi_1(\theta_1) \prod_{t=2}^{T} K_t(\theta_{t-1}, \theta_t) d\theta_{1:T-1}$$

is usually impossible to compute and therefore we can not calculate the necessary importance weights. Additionally, this assumes we are able to sample from $\pi_1(\theta_1)$, which is not always the case. Alternatives attempt to approximate η_t pointwise when possible, but the computation of these algorithms is in $O(N^2)$ (Del Moral et al., 2006).

The central idea of SMC Samplers (Del Moral et al., 2006) is to employ an auxiliary backward kernel with density $L_{t-1}(\theta_t, \theta_{t-1})$ to get around this intractible integral. This backward kernel relates to a time-reversed SMC sampler giving the same marginal distribution as the forward SMC sampler induced by K_t. The backward kernel is essentially arbitrary, but should be optimized to minimize the variance of the importance weights. Del Moral et al. (2006) prove that the sequence of backward kernels minimizing the variance of the importance weights is, for any t, $L_{t-1}^{\text{opt}}(\theta_t, \theta_{t-1}) = \eta_{t-1}(\theta_{t-1}) K_t(\theta_{t-1}, \theta_t) / \eta_t(\theta_t)$. However, it is typically impossible to use this optimal kernel since it relies on intractable marginals. Thus, we should select a backward kernel that approximates this optimal kernel. Del Moral et al. (2006) give two suboptimal backward kernels to approximate L_{t-1}^{opt}

which result in incremental weights

$$w_t^a(\theta_{t-1}, \theta_t) = \frac{\pi_t(\theta_t)}{\int \pi_{t-1}(\theta_{t-1}) K_t(\theta_{t-1}, \theta_t) d\theta_{t-1}} \tag{2.3a}$$

$$w_t^b(\theta_{t-1}, \theta_t) = \frac{\pi_t(\theta_{t-1})}{\pi_{t-1}(\theta_{t-1})}. \tag{2.3b}$$

These incremental weights are then multiplied by the weights at the previous time and normalized to sum to 1. We note that the suboptimal kernel resulting in (2.3b) is actually an approximation of that resulting in (2.3a), and coincidentally has the same form as (2.2), the reweighting mechanism for importance sampling. In this manner the first kernel should perform better, particularly when successive distributions are considerably different (Del Moral et al., 2006). Although the weights (2.3a) are a better approximation of the optimal backward kernel weights, the second kernel is convenient since the resulting incremental weights (2.3b) do not depend on the position of the moved particles θ_t and hence we are able to reweight the particles prior to moving them. We include the incremental weight (2.3a) because, when K_t is a Gibbs kernel moving one component at a time, it simplifies to $\pi_t(\theta_{t-1,-k})/\pi_{t-1}(\theta_{t-1,-k})$ where k is the index of the component being moved by the Gibbs sampler and $\theta_{t-1,-k}$ is the particle excluding the k^{th} component. By a simple Rao-Blackwell argument it can be seen that this choice, by conditioning on the variable being moved, results in reduced variance of the importance weights compared to (2.3b).

2.2.1 An Efficient SMC Algorithm

Now that we have described some components of SMC methodology, we proceed to develop an efficient algorithm for performing prior sensitivity and cross-validation. The basic idea of our algorithm is to first reweight the particles $\{W_{t-1}^{(i)}, \theta_{t-1}^{(i)}\}$, $i = 1, ..., N$ such that they are approximately distributed as π_t. If the variance of the weights is large, we then resample the particles with probabilities proportional to their weights, giving us a set of N equally weighted particles (including some duplicates). After resampling we move the particles with a kernel of invariant distribution π_t, which creates diversity in the particles. Our algorithm relates closely to resample-move algorithms (Gilks and Berzuini, 2001; Chopin, 2002), although our formulation is more general and allows for the use of a variety of suboptimal backward kernels and corresponding weights.

Moving the particles at each time step is not particularly efficient. For example, if two successive distributions in the sequence are identical, we

are wasting our time by moving the particles. If successive distributions are similar but not necessarily identical, to save computational time we can simply copy forward the particles at time $t-1$ and reweight them with the importance sampling weights (2.2). Deciding when to move particles may be done dynamically or deterministically. A dynamic scheme would move the particles whenever the variance of the weights becomes too large (usually measured by the effective sample size (ESS), $(\sum_{i=1}^{N}(W_t^{(i)})^2)^{-1}$), whereas a deterministic scheme would move the particles every k^{th} time step for some integer k. Since the sequence of distributions will likely not change at a constant rate, it is better to use a dynamic scheme as this allows for little particle movement during parts of the sequence with little change and more movement in parts of the sequence where successive distributions vary more.

When the ESS drops below a specified threshold, we reweight the particles at time $t-1$ to be approximately distributed as π_t prior to moving them. The weights (2.3b) only depend on the particles at time $t-1$, so we can easily do this. In the case of a one at a time Gibbs sampler, we can also use the weights (2.3a). Because the unweighted particles at time t are not distributed according to π_t, we cannot simply move the particles without first taking their weights into consideration. Thus prior to moving the particles we must resample them such that $W_t^{(i)} = 1/N$ for $i = 1, \ldots, N$ and the particles' unweighted distribution is π_t. Resampling methods duplicate particles with large weights and remove particles with low weights. Specifically, we copy the i^{th} particle $N_t^{(i)}$ times such that $\sum_{i=1}^{N} N_t^{(i)} = N$ and $E(N_t^{(i)}) = NW_t^{(i)}$ where $W_t^{(i)}$ are the normalized importance weights. Lastly, all of the resampled particles are assigned equal weight. The simplest unbiased resampling method consists of sampling $N_t^{(i)}$ from a multinomial distribution with parameter $(N, \{W_t^{(i)}\})$. It should be noted that more sophisticated resampling schemes, such as residual resampling (Liu, 2001) and stratified resampling (Kitigawa, 1996) exist, resulting in reduced variance of $N_t^{(i)}$ relative to multinomial resampling. After the particles are resampled, we can move them with the kernel K_t.

An efficient SMC algorithm which may be used to perform prior sensitivity and cross-validation is therefore:

for $t = 1$ **do**
  Obtain N weighted samples $\theta_1^{(i)}$ from π_1 (directly, MCMC, etc.)
end
for $t = 2, \ldots, T$ **do**
  Copy $\theta_{t-1}^{(i)}$ to $\theta_*^{(i)}$ and calculate weights $W_*^{(i)}$ according to (2.2)
  if $ESS(\theta_*) > c$ **then**
    Copy $\left(\theta_*^{(i)}, W_*^{(i)}\right)$ to $\left(\theta_t^{(i)}, W_t^{(i)}\right)$
  else
    Reweight: Calculate weights according to
    $W_t \propto W_{t-1} \times w_t(\theta_{t-1}, \theta_t)$ where $w_t(\theta_{t-1}, \theta_t)$ is either given by (3a) or (3b)
    Resample: Resample particles according to above weights. Set all weights to $1/N$
    Move: Move particles with Markov kernel of invariant distribution π_t
  end
end
note 1: If a backward kernel is chosen such that the incremental weights depend on the position of the moved particle $\theta_t^{(i)}$, the reweight step comes after the move step and resampling is performed with the weights $\theta_{t-1}^{(i)}$.
note 2: c is a user-specified threshold on the effective sample size.

2.3 Prior Sensitivity

In the case of prior sensitivity we are interested in approximating the posterior distribution of some variable(s) θ given the data D, symbolically notated as $\pi(\theta|D) \propto f(D|\theta) \cdot \nu(\theta)$ where $f(D|\theta)$ and $\nu(\theta)$ are the likelihood and the prior distribution of θ, respectively. Here the notation $\nu(\theta)$ is used to differentiate the prior from the posterior distribution $\pi_t(\theta)$, allowing for the omittance of dependencies. This prior sensitivity framework has been studied in a closed-form setting (Gustafson and Wasserman, 1995; Gustafson, 1996), but situations requiring Monte Carlo methods have received less attention. It is worth noting that only the prior distribution changes between successive distributions (the likelihood remains the same). Thus when we reweight particles to approximate the sequence of posterior distributions for

θ, the weights (2.2) depend solely on the prior distribution,

$$W_t^{(i)} \propto W_{t-1}^{(i)} \times \frac{f\left(D|\theta_{t-1}^{(i)}\right) \cdot \nu_t\left(\theta_{t-1}^{(i)}\right)}{f\left(D|\theta_{t-1}^{(i)}\right) \cdot \nu_{t-1}\left(\theta_{t-1}^{(i)}\right)}$$
$$\propto W_{t-1}^{(i)} \times \frac{\nu_t\left(\theta_{t-1}^{(i)}\right)}{\nu_{t-1}\left(\theta_{t-1}^{(i)}\right)} \tag{2.4}$$

where $\theta_t^{(i)}$ is the i^{th} particle sampled at time t and $\nu_t(\theta_t^{(i)})$ is the t^{th} prior distribution evaluated at the point $\theta_t^{(i)}$. If the ESS falls below a given threshold at time t (notated as c in algorithm pseudocode), we resample and move, otherwise we simply reweight. Conveniently, resampling and moving using (2.3b) and reweighting using (2.2) both result in the same weight mechanism (2.4). In a later example we will also employ the weights (2.3a), which have reduced variance relative to (2.3b).

2.3.1 Regularization Path Plots

Consider the regression model with response vector $\mathbf{y} = (y_1, \ldots, y_n)^T$ and model matrix $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_p})$ where $\mathbf{x_j} = (x_{1j}, \ldots, x_{nj})^T, j = 1, \ldots, p$ are the column vectors of predictors (including the unit intercept vector). For clarity of presentation we present the model with a continuous response; however, it is simple to extend to binary responses (Albert and Chib, 1993). We use the prostate data of Stamey et al. (1989) which has eight predictors and a response (logarithm of prostate-specific antigen) with likelihood

$$\mathbf{y}|\mu, \mathbf{X}, \boldsymbol{\beta}, \sigma^2 \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n). \tag{2.5}$$

Using a double exponential prior distribution with parameter λ on the regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, the corresponding posterior distribution is proportional to (2.1). We see from the form of this posterior distribution that if $\lambda = 0$ the MAP estimate of $\boldsymbol{\beta}$ will correspond to the least squares solution. However, as λ increases there will be shrinkage on $\boldsymbol{\beta}$ which may be displayed using a regularization path plot. Because the shrinkage as λ varies is nonlinear, we set a schedule $\lambda_t = e^{t/20}$, $t = 1, \ldots, 100$. We create a "gold standard" Bayesian Lasso regularization path plot for this data by running MCMC with a Markov chain of length 50,000 at each level of λ and plotting the posterior mean of the resulting regression coefficients (Figure 2.1). It should be noted that the creation of this plot took over 5 hours.

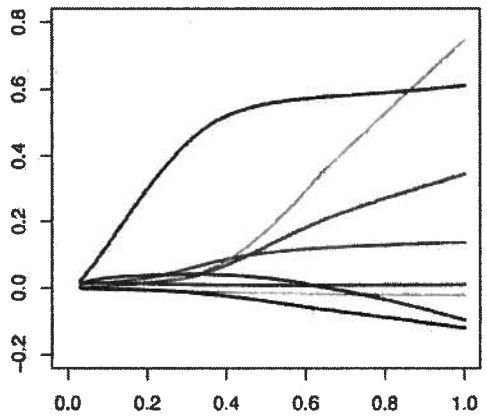


Figure 2.1: Regularization Path Plots: The gold standard
The plot is of standardized coefficients β_j vs. $|\boldsymbol{\beta}|_1/\max(|\boldsymbol{\beta}|_1)$.

Since the idea is to create these plots quickly for exploratory analysis, we will compare our SMC-based method to MCMC with both constrained to work in 5 minutes (+/− 5 seconds), and both using the same Markov kernel. In order to perform MCMC in our time frame of 5 minutes, the Markov chain had a length of 1200 for each of the 100 levels of λ. The mean of each resulting posterior distribution was used to plot the regularization path plots in Figure 2.2(a). In comparison, to run in 5 minutes our SMC algorithm used $N = 4200$ particles and resampled and moved the particles when the ESS dropped below $c = \frac{2N}{3} = 2800$ (Figure 2.2(b)). For the sake of time comparisons, all computations here and later were performed on a Power Mac G5 with dual 2.7 GHz PowerPC processors. We see from these plots that both methods capture the key features of the regularization path plot as shown by the gold standard: every one of the variables has the correct path. The methods vary, however, in the amount of noise. We see much more variability using MCMC compared to SMC. This is due to SMC being able to use many more particles since it is able to save time by borrowing information from previous distributions. To be specific, the SMC algorithm in this context had to resample and move the particles only 25 times in the entire sequence of 100 distributions. The remainder of the time our algorithm

simply reweighted the particles, which is computationally inexpensive. It is worth noting that, because of this, adding more incremental distributions in the sequence will have little effect on the computational time of the SMC algorithm, unlike MCMC-based strategies, which would approximate each new distribution with a new Markov chain. In addition, we attempted to make these plots using importance sampling, reweighting (and not moving) particles from π_1 to approximate later distributions. However, the weights became degenerate, with all of the weights eventually focussing on one particle with standardized L-1 norm of 0.8. Specifically, all but one of the weights had values near zero, and the one particle with positive weight had standardized L-1 norm of 0.8. Thus importance sampling was only able to create roughly 1/5 of the entire plot, and hence is clearly not a candidate methodology for creating these plots. We will see later that in many such situations importance sampling fails, even with large amounts of particles.

2.3.2 Variable Selection Using g-Priors

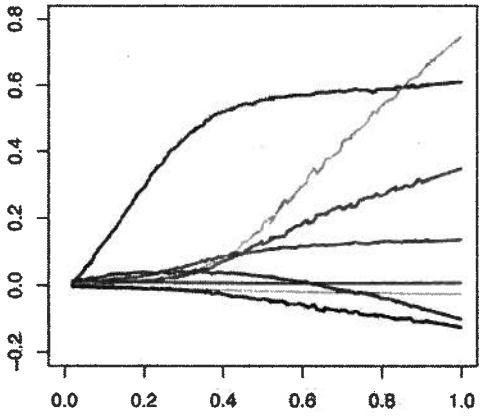
Consider the normal likelihood set-up (2.5). Now, however, with an eye towards model selection, we introduce the binary indicator variable $\gamma \in \{0,1\}^p$, where $\gamma_j = 1$ means the variable $\boldsymbol{x}_j$ is included in the model. Thus γ can describe all of the 2^p possible models. Following the notation of Marin and Robert (2007), we use $q_\gamma = \mathbf{1}_n^T \gamma$ as a counter for the number of variables in the model. If $\mathbf{X}_\gamma$ is the model matrix which excludes all $\boldsymbol{x}_j$'s if $\gamma_j = 0$, we can employ the following prior distributions for $\boldsymbol{\beta}$ and σ^2 (Zellner, 1986; Marin and Robert, 2007):

$$\pi(\boldsymbol{\beta}_\gamma, \sigma^2|\gamma) \propto (\sigma^2)^{-(q_\gamma+1)/2-1} \exp\left[-\frac{1}{2g\sigma^2}\boldsymbol{\beta}_\gamma^T(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)\boldsymbol{\beta}_\gamma\right].$$

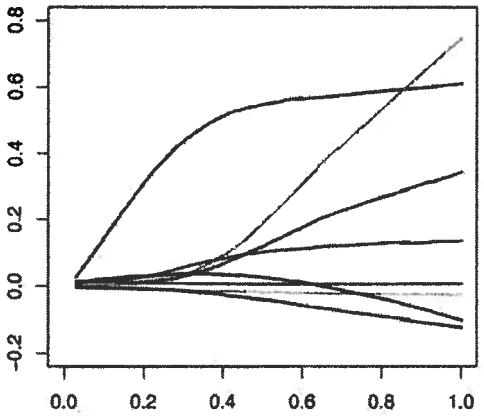
From this it is straightforward to show that the posterior density for γ is thus

$$\pi(\gamma|\mathbf{y},\mathbf{X}) \propto (g+1)^{-(q_\gamma+1)/2}\left[\mathbf{y}^T\mathbf{y} - \frac{g}{g+1}\mathbf{y}^T\mathbf{X}_\gamma(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1}\mathbf{X}_\gamma\mathbf{y}\right]^{-n/2}. \tag{2.6}$$

We perform model selection on the pollution data set of McDonald and Schwing (1973), in which mortality rate is compared against 15 pollution related variables in 60 metropolitan areas. The 15 independent variables include, for instance, mean annual precipitation, population per household, and average annual relative humidity. The response variable $\mathbf{y}$ is the age-adjusted mortality rate in the given metropolitan area. We seek to perform



(a) MCMC with 1200 samples (5 minutes)



(b) SMC with 4200 samples (5 minutes)

Figure 2.2: Regularization Path Plots: Plots using MCMC and SMC for fixed computational time of 5 minutes

The plots are of standardized coefficients β_j vs. $|\boldsymbol{\beta}|_1/max(|\boldsymbol{\beta}|_1)$.

variable selection to narrow down the number of independent variables which best predict the response. With 15 variables, calculating the posterior probabilities of the over 30,000 models exactly is possible but time-consuming. We have chosen this size of data set to allow for a benchmark from which we can compare MCMC to SMC.

Our goal is to see how the explanatory variables change as we vary the prior distribution parameter g. In other words, we are interested in seeing how robust the variable selection method is to changes in the setting of g. Our goal is to perform the variable selection for 100 levels of g for schedule $g = e^{t/10}$, $t = 1, \ldots, 100$. We use a Gibbs sampler strategy to compare the SMC-based algorithm to brute-force MCMC, benchmarked against the exact solution obtained from (2.6), in which β_γ and σ^2 are integrated out. Specifically, we update γ one component at a time. The incremental weight ratio (2.3b) will be the ratio of the posterior distribution (2.6) evaluated on the complete data at successive levels of g. In addition, we are able to use the weights (2.3a), which corresponds to the ratio of the posterior distribution (2.6) evaluated on all of the data, excluding the variable that is being moved by the Gibbs sampler.

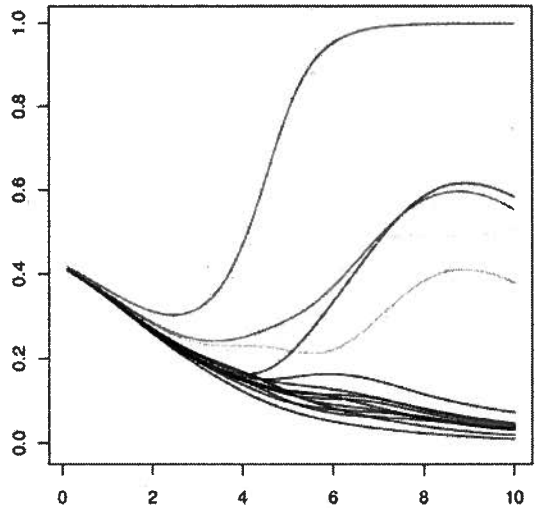
In order to see our desired result, we use (2.6) to plot the exact marginal probabilities as well as some sample model probabilities for various levels of g (Figures 2.3(a) and 2.3(b)). This process took slightly over 8 hours, and hence we would like to find a faster method. We constrain both stochastic algorithms to run in 30 minutes (+/- 1 minute). As a result the MCMC algorithm uses a Markov chain of length 10,000 and the SMC algorithm uses 18,000 particles. We plot the resulting posterior marginal probabilities for each algorithm in Figures 2.4(a) and 2.4(b), respectively. First impression shows that the plot created using MCMC has much more variability. However, the smoothness in the SMC algorithm is not a result of perfect accuracy of the method, but rather only smoothness of the reweighting mechanism (2.2). Because of this, if the SMC does poorly during times of particle movement, the subsequent reweighted approximations will also be inaccurate. To ensure this is not the case and verify that SMC is indeed outperforming MCMC, we look at the average absolute error of the marginal probabilities (at 100 levels of λ and for 15 variables). We find the average absolute error in the marginal probabilities using MCMC is 0.0292 whereas with SMC it is only 0.0187. In addition, their respective maximum absolute errors were 0.24 and 0.08, respectively. In fact 30 runs of the algorithms resulted in similar results, with SMC consistently outperforming MCMC. From this we see that SMC is indeed providing a better approximation of the true marginal probabilities.

What then may be taken from these marginal probability plots? When performing simple forward selection regression, the variables 1, 2, 6, 9, and 14 are chosen. Slightly different results come from doing backward selection; in particular variables 1 and 14 are replaced by variables 12 and 13. The LASSO solution (using 5-fold cross-validation) is the same as the forward solution with the additional variables 7 and 8. In addition, the LASSO solution contains some shrinkage on the regression coefficients (see example 2.3.1). Using g-Priors the variables that clearly stand out (see Figure 2.3(a)) are 1, 2, 6, 9, and 14. Thus the g-Prior solution taken from the plot corresponds to the forward selection model. Also, for a given g, say $g = e^9$, the plot obtained with SMC shows the correct top 4 variables for inclusion, whereas the variability from the MCMC-based plot makes it impossible to do so.

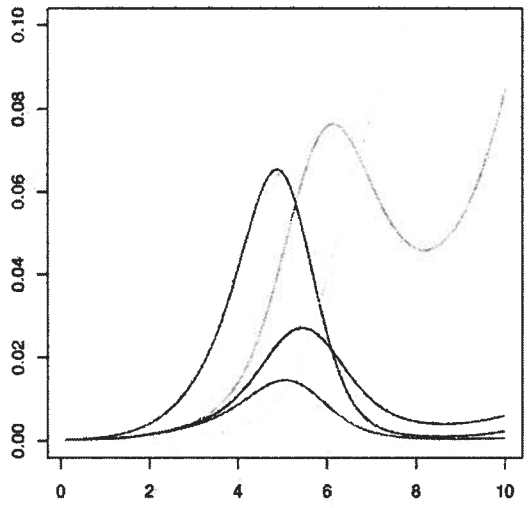
2.4 Cross-Validation

We focus on leave-s-out cross-validation, which is the case when the testing set consists of s observations. Continuing in the linear regression framework, let $\mathbf{X}_{\setminus S}$ and $\mathbf{y}_{\setminus S}$ be the model matrix and response vector excluding the subset S of observations (of size s). We are interesting in a collection of T model parameter (typically prior distribution parameter) settings resulting in posterior densities $\pi_t(\theta|\mathbf{X}_{\setminus S}, \mathbf{y}_{\setminus S})$ for $t = 1, \ldots, T$. Once we have approximations of all T posterior densities, we select the model parameter settings which result in the best prediction of $\mathbf{y}_S$ using $\mathbf{X}_S$. To find the sequence of distributions $\pi_t(\theta|\mathbf{X}_{\setminus S}, \mathbf{y}_{\setminus S})$, $t = 1, \ldots, T$, the same SMC-based algorithm proposed for prior sensitivity is applicable. Specifically, once we have obtained a Monte Carlo approximation of $\pi_1(\theta|\mathbf{X}_{\setminus S}, \mathbf{y}_{\setminus S})$, we can transition to the remainder of the distributions $\pi_t(\theta|\mathbf{X}_{\setminus S}, \mathbf{y}_{\setminus S})$, $t = 2, \ldots, T$ using SMC.

In addition to quickly evaluating the model for a variety of settings on the training set, SMC also provides a tool for switching the training/testing set without fully re-approximating the posterior densities. Specifically, suppose we have a testing set S_1, and using SMC we find approximations of $\pi_t(\theta|\mathbf{X}_{\setminus S_1}, \mathbf{y}_{\setminus S_1})$, $t = 1, \ldots, T$, each of which are tested for prediction performance on the subset S_1. However, typically we are interested in performing cross-validation for a variety of different splits of the data into training and testing sets. Thus, we will now want a new testing set S_2 and find approximations of $\pi_t(\theta|\mathbf{X}_{\setminus S_2}, \mathbf{y}_{\setminus S_2})$, $t = 1, \ldots, T$. The obvious way to accomplish this is to start fresh by approximating $\pi_1(\theta|\mathbf{X}_{\setminus S_2}, \mathbf{y}_{\setminus S_2})$ with MCMC and proceeding to approximate the remainder of the distributions using SMC. However,

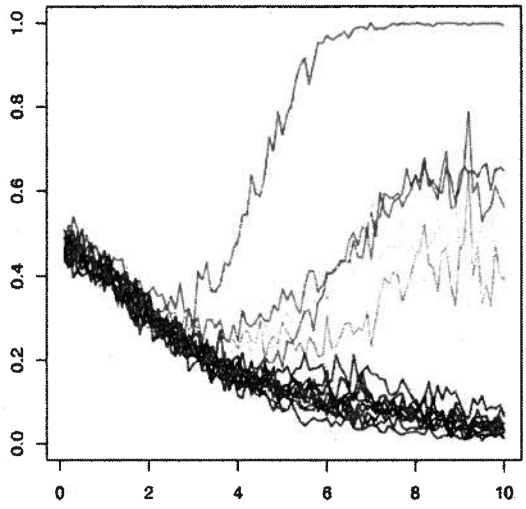


(a) Posterior Marginal Probabilities: Green=$\mathbf{X}_1$, Blue=$\mathbf{X}_2$, Yellow=$\mathbf{X}_6$, Red=$\mathbf{X}_9$, Purple=$\mathbf{X}_{14}$

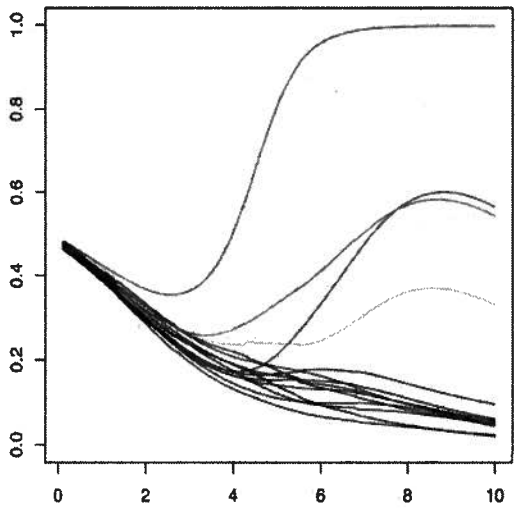


(b) Posterior Model probabilties: Purple=1,2,4,5,9, Red=1,2,4,5, Blue=1,9, Green=9, Yellow=6

Figure 2.3: Exact Marginal and Model probabilities for variable selection using g-Priors as a function of $log(g)$
Plot (a) highlights several variables ($\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_6, \mathbf{X}_9, \mathbf{X}_{14}$) which show high marginal probabilities of inclusion. Plot (b) shows the posterior probabilities of 5 models chosen to highlight the effect of g on model size.



(a) Posterior Marginal Probabilities: MCMC with 10000 samples (30 minutes)



(b) Posterior Marginal Probabilities: SMC with 18000 samples (30 minutes)

Figure 2.4: Approximate Marginal and Model probabilities for variable selection using g-Priors as a function of $log(g)$

Plots (a) and (b) compare MCMC to SMC's performance.

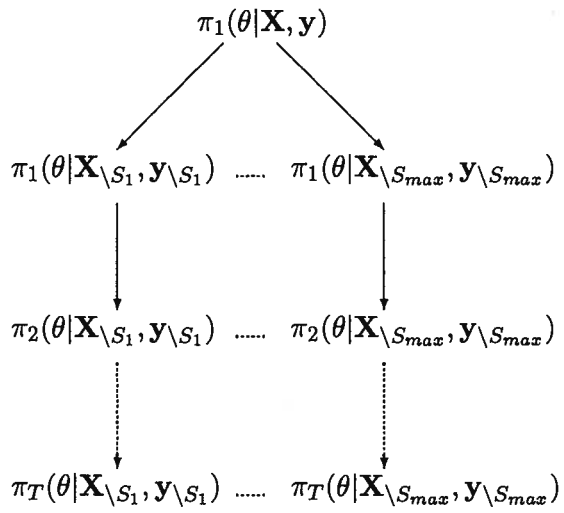


Figure 2.5: Diagram of cross-validation process
Each arrow represents transitioning using SMC.

we can be a bit more clever than this, recognizing that $\pi_1(\theta|\mathbf{X}_{\setminus S_1},\mathbf{y}_{\setminus S_1})$ and $\pi_1(\theta|\mathbf{X}_{\setminus S_2},\mathbf{y}_{\setminus S_2})$ are related (Alqallaf and Gustafson, 2001; Bhattacharya and Haslett, 2007).

Successive splits of the data into training and testing sets should give similar model settings. Therefore, we first build the model for a given parameter setting on the full data set using SMC, resulting in an approximation of $\pi_1(\theta|\mathbf{X},\mathbf{y})$. Then instead of using MCMC to get approximations of $\pi_1(\theta|\mathbf{X}_{\setminus S},\mathbf{y}_{\setminus S})$ for different $S \in \{S_1,\ldots,S_{max}\}$, we can build a sequence of distributions $(\pi_1(\theta|\mathbf{X},\mathbf{y}))^{1-\gamma}(\pi_1(\theta|\mathbf{X}_{\setminus S},\mathbf{y}_{\setminus S}))^{\gamma}$ for an increasing temperature $\gamma = 0,\epsilon,2\epsilon,\ldots,1-\epsilon,1$ which will allow us to transition to the case-deletion posteriors. The process is illustrated in Figure 2.5. The case of $\gamma = 0,1$ with no movement step corresponds to basic case-deletion importance sampling as developed in Peruggia (1997). Although case-deletion importance sampling has been demonstrated to achieve up to 90% cost savings in some circumstances (Alqallaf and Gustafson, 2001), the problem of degeneracy still makes importance sampling fail in many situations (Peruggia, 1997; Epifani et al., 2005).

Let $\Theta = (\boldsymbol{\beta},\sigma^2)$. The posterior distribution $\pi(\Theta)$ of Θ is proportional to $q(\Theta) = f(\mathbf{y}|\boldsymbol{\beta},\sigma^2)\times\pi(\boldsymbol{\beta})\times\pi(\sigma^2)$. Assume we collect samples from the distribution $\pi(\Theta)$. We are interested in reweighting these samples such that they come from the distribution attained by removing the set S. The modified

likelihood and posterior for this case-deletion scenario are, respectively

$$f_{\backslash S}(\mathbf{y}|\boldsymbol{\beta},\sigma^2) = (\sigma^2)^{-(n-s)/2}\exp\{-\frac{1}{2\sigma^2}\left[(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - (\mathbf{y}_S - \mathbf{X}_S^T\boldsymbol{\beta})^T((\mathbf{y}_S - \mathbf{X}_S^T\boldsymbol{\beta}))\right]\}$$
$$q_{\backslash S}(\Theta) = f_{\backslash S}(\mathbf{y}|\boldsymbol{\beta},\sigma^2) \times \pi(\boldsymbol{\beta}) \times \pi(\sigma^2)$$

We assume that the prior distributions for $\boldsymbol{\beta}$ and σ^2 are proper and independent. Epifani et al. (2005) show that if the weights $w_{\backslash S}(\Theta) = q_{\backslash S}(\Theta)/q(\Theta)$ are used to move to the case-deletion posterior directly, then the r^{th} moment of these weights is finite if and only if all of the following conditions hold:

$$\begin{array}{lcl} a) & \lambda_H & < 1/r \\ b) & n - rs & > 1 \\ c) & \text{RSS}^*_{\backslash S}(r) & > 0 \end{array}$$

where λ_H is the largest eigenvalue of the matrix $H_S = \mathbf{X}_S^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_S$ and $\text{RSS}^*_{\backslash S}(r) = \text{RSS} - re_S^T(\mathbf{I} - rH_S)^{-1}e_S$ where $e_S = y_S - \mathbf{X}_S^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ and RSS denotes the residual sum of squares of the least squares fit of the full data set. This result should not be taken lightly: as Geweke (1989) points out, if the 2nd moment does not exist, the importance sampling estimator will follow neither a $n^{1/2}$ asymptotic nor a central limit theorem. (a) states that if the leverage of the deleted observations is too large, then the importance weights will have infinite variance. (b) gives a condition relating sample size to the allowable test set size s. (c) says that if the influence of the deleted observation is large relative to RSS, then the importance weights will have infinite variance. We show here how using a sequence of artificial intermediate distributions with SMC can help to mitigate this problem.

We introduce a sequence of distributions

$$q_\gamma(\Theta) \propto (q(\Theta))^{1-\gamma}(q_{\backslash S}(\Theta))^\gamma$$

where $\gamma = 0, \epsilon, 2\epsilon, \ldots, 1-\epsilon, 1$ to move from $q(\Theta) = q_0(\Theta)$ to $q_{\backslash S}(\Theta) = q_1(\Theta)$. At a given step $\gamma = \gamma^*$ in the sequence, the successive importance weights appearing in the SMC algorithm to move to the next step $\gamma^* + \epsilon$ are

$$\begin{aligned} w_{\backslash S,\gamma^*}(\Theta) &= \frac{(q(\Theta))^{1-\gamma^*-\epsilon}(q_{\backslash S}(\Theta))^{\gamma^*+\epsilon}}{(q(\Theta))^{1-\gamma^*}(q_{\backslash S}(\Theta))^{\gamma^*}} \\ &= \left(\frac{q_{\backslash S}(\Theta)}{q(\Theta)}\right)^\epsilon \end{aligned}$$

Theorem 1. *Provided that* $RSS*_{\setminus S}(1) > 0$ *and the prior distributions for* β *and* σ^2 *are proper and independent, a sequence of distributions proportional to* $\{(q(\Theta))^{1-\gamma}(q_{\setminus S}(\Theta))^{\gamma}; \gamma = 0, \epsilon, 2\epsilon, \ldots, 1-\epsilon, 1\}$ *may be constructed to move from* $q(\Theta)$ *to* $q_{\setminus S}(\Theta)$ *such that the importance weights* $w_{\setminus S,\gamma}(\Theta)$ *for each successive step have a finite* r^{th} *moment under* $q_{\gamma}(\Theta)$ *provided*

$$\epsilon < \frac{\alpha - 1}{r - 1} \tag{2.7}$$

where $\alpha > 1$ *is chosen to satisfy*

$$\lambda_H < 1/\alpha \tag{2.8a}$$

$$n - \alpha s > 2 \tag{2.8b}$$

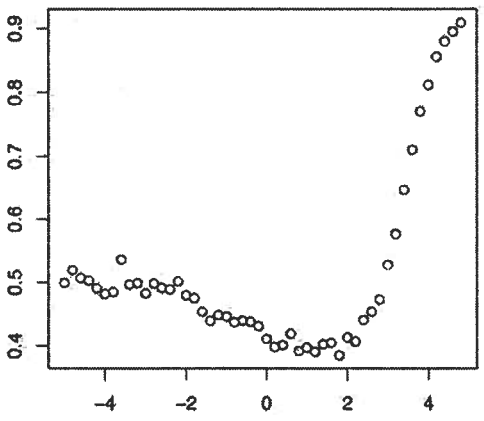
$$RSS*_{\setminus S}(\alpha) > 0 \tag{2.8c}$$

The proof may be found in the appendix. The provision that $RSS*_{\setminus S}(1) > 0$ is very reasonable, and states that the least squares fit of the full data must not fit the training set perfectly. Note also that we find α for each subset S. Thus we may use the largest allowable step size ϵ in (2.7) for each subset S, maximizing the algorithm's efficiency. While this result is not sufficient to establish that the variance of SMC estimates are finite for a finite number N of particles, it can be used to upper bound the asymptotic variance of SMC estimates under additional mild regularity mixing conditions on the MCMC kernels; see (Chopin, 2004) and (Jasra and Doucet, 2008) for similar ideas.

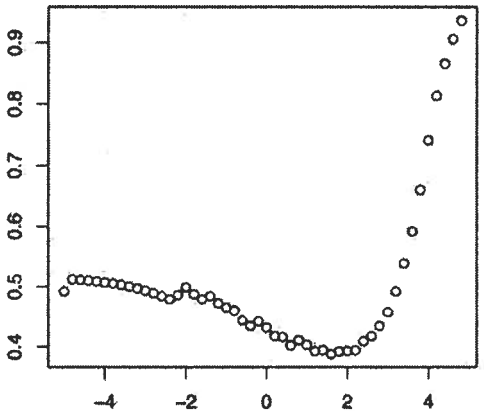
2.4.1 Application to Bayesian Penalized Regression

To demonstrate the strength of SMC applied to cross-validation, we use it to select the parameter λ of the Bayesian Lasso (2.1). For brevity, we reuse the pollution data set (McDonald and Schwing, 1973) of section 2.3.2, selecting the parameter λ using leave-one-out cross-validation. Firstly, it is worth pointing out that importance sampling will fail in this situation, as $\lambda_H > 1/2$ on 6 of the 60 observations in this data set, and hence the sufficient conditions to ensure finite variance are not satisfied. Using a sequence of intermediate distributions, we find that the largest α satisfying (8) equals 1.103, or, to ensure a finite second moment, $\epsilon < \frac{\alpha-1}{r-1} = .103$. Thus, the largest sequence of distributions was of length 10. For most variables $\alpha > 2$, which for $r = 2$ is equivalent to importance sampling. Thus SMC does not waste time transitioning to case-deleted posteriors if importance sampling will suffice.

We use a Gibbs sampler to approximate the posterior distribution of (β, σ^2) for $\lambda = e^{-5}$ on the full data set and then use SMC to move to



(a) Cross-validation error as a function of $\log(\lambda)$ using MCMC with 60 samples (10 minutes)



(b) Cross-validation error as a function of $\log(\lambda)$ using SMC with 100 samples (10 minutes)

Figure 2.6: Plots of cross-validation error as a function of $\log(\lambda)$

the case-deletion posterior distributions by creating a sequence of auxiliary distributions as described above. For each different case-deletion we then use SMC to find approximations of the posterior for schedule $\lambda = e^{t/5}$, $t = -25, \ldots, 25$. Plotting the cross-validation errors as a function of λ using MCMC with a Markov chain of length 10,000 (not shown) we observe that the average squared loss $\frac{1}{21}\sum_{k=1}^{21}(\mathbf{y}_k - \boldsymbol{x}_k\boldsymbol{\beta})^2$ is a smooth function in λ with a small bump at $\lambda = e^2$ and minimum near $e^{3/2}$. Thus to minimize prediction error (at least in terms of the squared loss) we should set $\lambda = e^{3/2}$. To perform this task in a time-restricted manner we constrained both MCMC and SMC algorithms to work in 10 minutes (+/- 30 seconds). Figures 2.6(a) and 2.6(b) are the resulting plots. The reduced variability of the SMC-based plot allows us to make more accurate conclusions. For instance, it is clear in the plot obtained with SMC (Figure 2.6(b)) that the minimum error lies somewhere around $\lambda = e^{3/2}$, whereas from the MCMC plot (Figure 2.6(a)) it could be anywhere between 1 and e^2.

2.5 Extensions and Conclusions

In our presentation of the algorithm, a fixed sequence of distributions $\pi_t(\theta)$, $t = 1, 2, 3, .., T$ is used. However, it is also possible to determine the sequence of distributions automatically such that successive distributions are a fixed distance apart, as measured by ESS. For instance, assume we are interested in $\pi_t(\theta) = \pi(\theta|\lambda_t)$ where λ_t is a scalar parameter and we have a Monte Carlo approximation of $\pi(\theta|\lambda_{t-1})$ for an arbitrary t, namely $\{W_{t-1}^{(i)}, \theta_{t-1}^{(i)}\}$, $i = 1, \ldots, N$. We may set λ_t to ensure that $ESS = c$ for a constant c by solving

$$c = \sum_{i=1}^{N} \left((W_t^{(i)})^2 \right)^{-1}$$

where $W_t^{(i)}$ is given by (2.2). This may be solved numerically or in closed-form, if possible. This technique would be beneficial in situations where little or nothing is known about the sequence of distributions, and hence it would be nice to automatically create the sequence.

All our examples have considered a sequence of distributions parameterized by a scalar parameter for which the definition of the sequence of target distributions is very intuitive. If we are interested in dealing with multivariate parameters then the algorithm may be adapted by, for instance, creating a grid (or hyper-grid) of distributions. SMC may be used to work across

each dimension in succession. It is worth noting that the complexity of the algorithm scales exponentially with dimension, although MCMC does as well.

While we have given two choices of incremental weights, (2.3a) and (2.3b), many other choices are available (Del Moral et al., 2006). In situations where the weights are dependent on the position of the moved particle, such as with (2.3a), auxiliary particle techniques may be used (Pitt and Shephard, 1999; Johansen and Whiteley, 2008). Specifically, we reweight the particles with an approximation of the weight of interest (for instance, (2.3a)) which is only dependent on the particles at time $t-1$, using $W_{\text{temp}}^{(i)} \propto W_{t-1}^{(i)} \times W_*^{(i)}$ where $W_*^{(i)}$ is the approximation of the incremental weight. After we have resampled and moved the particles we then compensate for this approximation using $W_t^{(i)} = \frac{W_{\text{true}}^{(i)}}{W_*^{(i)}} \times W_{\text{temp}}^{(i)}$.

We have seen that by adapting importance sampling to move particles between successive distributions, SMC drastically limits the problem of importance sampling degeneracy. By using a resample-move type algorithm, we are able to perform prior sensitivity and cross-validation in a computationally feasible manner while avoiding the fore-mentioned pitfalls of importance sampling. We have shown the SMC algorithm to be considerably more efficient than existing methods based on reiterative MCMC approximations. In this way regularization path plots and other sensitivity analysis problems can be studied in the context of the full posterior distribution instead of a few summary statistics. In addition, SMC provides a tool for naturally performing cross-validation, and in fact guarantees finite case-deletion weights under much less stringent conditions than importance sampling. In addition, through the importance weights, SMC provides a measure of the distance between distributions, and hence gives a way to select a subset of distributions of interest for exploratory or other purposes.

2.6 References

Albert, J.H. and Chib, S. (1993). "Bayesian Analysis of Binary and Polytomous Response Data." *Journal of the American Statistical Association.* 88:669-679.

Alqallaf, F. and Gustafson, P. (2001). "On Cross-validation of Bayesian Models." *Canadian Journal of Statistics.* 29:333-340.

Besag, J., Green, P., Higdon, D., Mengersen, K. (1995) "Bayesian Computation and Stochastic Systems (with discussion)." *Journal of the Ameri-*

can Statistical Association. 95:1127-1142.

Bhattacharya, S., and Haslett, J. (2007). "Importance Re-sampling MCMC for Cross-Validation in Inverse Problems." *Bayesian Analysis.* 2:385-408.

Chopin, N. (2002). "A Sequential Particle Filter Method for Static Models." *Biometrika.* 89:539-552.

Chopin, N. (2004). "Central Limit Theorem for Sequential Monte Carlo Methods and Its Application to Bayesian Inference." *Annals of Statistics.* 32:2385-2411.

Del Moral, P., Doucet, A., Jasra, A. (2006). "Sequential Monte Carlo Samplers." *Journal of the Royal Statistical Society: Series B.* 68:411-436.

Doucet, A., Godsill, S., Andrieu, C. (2000). "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering." *Statistics and Computing.* 10:197-208

Doucet, A., de Freitas, N., Gordon, N.J. eds. (2001). *Sequential Monte Carlo Methods in Practice.* New York: Springer.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. (2004). "Least Angle Regression." *Annals of Statistics.* 32:407-499.

Epifani, I., MacEachern, S., Peruggia, M. (2005). "Case-Deletion Importance Sampling Estimators: Central Limit Theorems and Related Results." Technical Report No. 720, Department of Statistics, Ohio State University.

Geweke, J. (1989). "Bayesian Inference in Econometric Models Using Monte Carlo Integration." *Journal of the American Statistical Association,* 88:881-889.

Gilks, W.R. and Berzuini, C. (2001). "Following a Moving Target: Monte Carlo Inference for Dynamic Bayesian Models." *Journal of the Royal Statistical Society: Series B.* 63:127-146.

Gustafson, P. and Wasserman, L. (1995). "Local Sensitivity Diagnostics for Bayesian Inference." *Annals of Statistics.* 23:2153-2167.

Gustafson, P. (1996). "Local Sensitivity of Inferences to Prior Marginals." *Journal of the American Statistical Association.* 91:774-781.

Jasra, A. and Doucet, A. "Stability of Sequential Monte Carlo Samplers via the Foster-Lyapunov Condition", *Statistics and Probability Letters.* to appear.

Johansen, A., and Whiteley, N. (2008). "A Modern Perspective on Auxiliary Particle Filters." *Proceedings of Workshop on Inference and Estimation in Probabilistic Time Series Models. Isaac Newton Institute, June*

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983). "Optimization by Simulated Annealing." *Science.* 220:671-680.

Kitagawa, G. (1996). "Monte Carlo Filter and Smoother for Non-Gaussian, Non-linear State Space Models." *Journal of Computational and Graphical Statistics.* 5:1-25.

Liu, J.S. and Chen, R. (1998). "Sequential Monte Carlo Methods for Dynamic Systems." *Journal of the American Statistical Association.* 93:1032-1044.

Liu, J.S. (2001). *Monte Carlo Strategies in Scientific Computing* (2nd ed.), New York: Springer.

McDonald, G. and Schwing, R. (1973). "Instabilities of Regression Estimates Relating Air Pollution to Mortality." *Technometrics.* 15: 463-481.

Neal, R. (2001). "Annealed Importance Sampling." *Statistical Computing.* 11:125-139.

Park, T. and Casella, G. (2008). "The Bayesian Lasso." *Journal of the American Statistical Association.* 103:681-686.

Peruggia, M. (1997). "On the Variability of Case-Deletion Importance Sampling Weights in the Bayesian Linear Model." *Journal of the American Statistical Association.* 92:199-207.

Pitt, M.K. and Shephard, N. (1999). "Filtering Via Simulation: Auxiliary Particle Filters." *Journal of the American Statistical Association.* 94:590-591.

Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society: Series B.* 58:267-288.

Vidakovic, B. (1998). "Wavelet-Based Nonparametric Bayes Methods." in *Practical Nonparametric and Semiparametric Bayesian Statistics.* D. Dey, P. Muller, and D Sinha (eds.). New York: Springer, 133-256.

Zellner, A. (1986). "On Assessing Prior Distributions and Bayesian Regression Analysis with G-prior Distributions." *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti.* pp. 233-243.

Chapter 3

Structural Health Monitoring with Autoregressive Support Vector Machines

3.1 Introduction

[2]The extensive literature on structural health monitoring (SHM) has documented the critical importance of detecting damage in aerospace, civil, and mechanical engineering systems at the earliest possible time. For instance, airlines may be interested in maximizing the lifespan and reliability of their jet engines or governmental authorities might like to monitor the condition of bridges and other civil infrastructure in an effort to develop cost-effective lifecycle maintenance strategies. These examples indicate that the ability to efficiently and accurately monitor all types of structural systems is crucial for both economic and life-safety issues. One such monitoring technique is vibration-based damage detection, which is based on the principal that damage in a structure, such as a loosened connection or crack, will alter the dynamic response of that structure. There has been much recent work in this area; in particular, Doebling et al. (1998) and Sohn et al. (2004) present detailed reviews of vibration-based SHM. Because of random and systematic variability in experimentally measured dynamic response data, statistical approaches are necessary to ensure that changes in a structures measured dynamic response are a result of damage and not caused by operational and environmental variability. Although much of the vibration-based SHM literature focuses on deterministic methods for identifying damage from changes in dynamic system response, we will focus on approaches that follow a sta-

[2]A version of this chapter has been accepted for publication. Bornn, L., Farrar, C.R., Park, G., Farinholt, K. (2008). "Structural Health Monitoring with Autoregressive Support Vector Machines." *Journal of Vibration and Acoustics.*

tistical pattern recognition paradigm for SHM (Farrar and Worden, 2008). This paradigm consists of the four steps of 1. Operational evaluation, 2. Data acquisition, 3. Feature extraction, and 4. Statistical classification of features. The work presented herein focus on steps 3 and 4 of this paradigm.

One approach for performing SHM is to fit a time series predictive model such as an autoregressive (AR) model to each sensor output using data known to be acquired from the structure in its undamaged state. These models are then used to predict subsequent measured data, and the residuals (the difference between the model's prediction and the observed value) are the damage-sensitive feature that is used to check for anomalies. This process provides many estimates (one at each time step) of a single-dimension feature, which is advantageous for subsequent statistical classification. The logic behind this approach is that if the model fit to the undamaged sensor data no longer predicts the data subsequently obtained from the system (and hence the residuals are large and/or correlated), there has been some sort of change in the process underlying the generation of the data. This change is assumed to be caused by damage to the system. These linear time series models have been used in such a damage detection process that include applications to a wide range of structures and associated damage scenarios including cracking in concrete columns (Fugate et al., 2001; Sohn et al., 2000), loose connections in a bolted metallic frame structure (Allen et al., 2002) and damage to insulation on wiring (Clark, 2008). However, the linear nature of this modeling approach limits the scope of application and the ability to accurately assess the condition of systems that exhibit nonlinearity in their undamaged state. In this paper, we demonstrate how support vector machines (SVM) may be used to create a non-linear time series model that provides an alternative to these linear AR models.

Once a model has been chosen and the predictions from this model have been compared to actual sensor data, there are several statistical methods for analyzing the resulting residuals. Sequential hypothesis tests, such as the sequential probability ratio test (Allen et al., 2002), may be used to test for changes in the residuals. Alternatively, statistical process control procedures, typically in the form of control charts, may be used to indicate abnormalities in the residuals (Fugate et al., 2001). In addition, sliding window approaches look at the features of successive subsets of data to detect anomalies (e.g. Clark, 2008). For example, the sliding window approach of Ma and Perkins (2003) looks at thresholds for the residuals such that the probability of an undamaged residual exceeding this threshold is 5%. A subset of n consecutive data points are then checked, and large values of the number g of points exceeding the threshold indicate damage, where g has a

binomial distribution (i.e. $g \sim Bin(n, .05)$).

To date, most of these time series modeling approaches analyze data from one sensor at a time, and typically some sort of scheme is used to determine how many sensors need to indicate damage in order to trigger a system check (e.g. Herzog et al., 2005). As an alternative, in this paper we look at a statistically based method for combining multiple sensor output. From this combined output analysis, we can establish the existence of damage and also determine which sensors are contributing to the anomalous readings in an effort to locate the damage within the sensor network's spatial distribution. Previously Sohn et al. (2000) have used principal component analysis to combine data from an array of sensors, but this study only examined these combined data in an effort to establish the existence of damage.

We first present a summary of the SVM approach to nonlinear time series modeling. This procedure is illustrated on numerically generated data with artificial anomalies added to the baseline signal in an effort to simulate damage. This time series modeling approach is then compared to linear AR models. Next the SVM method is coupled with a statistical analysis procedure that combines modeling results from multiple sensors in an effort to both establish the existence and the location of the damage. This procedure is applied to data from a laboratory test structure with damage that results in local nonlinear system response.

3.2 SVM-based SHM

Existing methods for performing damage detection extract damage-sensitive features from data acquired on the undamaged system, and then use changes in those features as an indicator of damage. An AR model can be fit to the undamaged sensor output and the residuals from predictions of subsequent data using this baseline model are then monitored for statistically significant changes that are assumed to be caused by damage. Specifically, an AR model with p autoregressive terms, AR(p), applied to sensor k may be written as

$$x_t^k = \sum_{j=1}^{p} \beta_j^k \cdot x_{t-j}^k + \epsilon_t^k \tag{3.1}$$

where x_t^k is the representation of the measured signal at discrete times t from the k^{th} sensor, β_j^k are the AR coefficients or model parameters, and ϵ_t^k is an unobservable noise term. Thus an AR model works by fitting a simple linear model to each point with the previous p observed points as dependent

variables. Note that an n point time series will yield $n - p$ equations that can be used to generate a least square estimate of the AR coefficients or the Yule-Walker Method can be used to solve for the coefficients (Brockwell and Davis, 2001). Auto-regressive models work particularly well when modeling the response of linear, time-invariant systems. If the undamaged system is nonlinear, the AR process gives the best linear fit to the measured response, but there is no guarantee that this model will accurately predict responses obtained when the system is subjected to other inputs.

Because of the broad array of structural health monitoring problems, employing a linear model confines the scope of problems for which the AR methodology is appropriate. We thus seek to extend the fidelity of this general damage detection approach by employing a non-linear AR-type model based upon SVMs, which have seen widespread use in machine learning and statistical classification fields. To simplify future development, we denote the vector $\{x^k_{t-p}, \ldots, x^k_{t-1}\}$ as $\mathbf{x}^k_{t-p:t-1}$. SVMs have many features that make them a more appropriate choice for SHM based on time series analysis. With the right settings and appropriate training they are able to model any non-linear relationship between the current time point, x^k_t, and the p previous time points, $\mathbf{x}^k_{t-p:t-1}$, they are well suited for high-dimensional problems, and the methodology is easily generalized and highly adaptable. Although SVMs have been used for SHM before (e.g. Worden and Manson, 2007; Shimada et al., 2006; Bulut et al., 2005, Worden and Lane, 2001; Chattopadhyay et al., 2007), these approaches predominantly focus on one and two class SVMs, which are used for outlier detection and group classification, respectively. Our approach is unique in its combination of support vector regression, autoregressive techniques, and residual error analysis. Thus while earlier approaches look at classifying sections of the time-series response as damaged or undamaged directly (the dependent variable being a binary indicator), our methodology works by using support vector regression to model the raw time-series data, then subsequently predicting damage by monitoring the residuals of the model. We follow the development of SVMs for regression of Smola and Schlkopf (2004) and Ma and Perkins (2003).

First, assume we have data from a set of K sensors and we have measurements without damage for time $t = 1, \ldots, t_0$ (i.e. if there is damage, it occurs after time t_0). Next we must decide the order p of our model. There are many methods for selecting p, such as partial autocorrelation or the Akaike Information Criterion (AIC), which are discussed in more detail in Fugate et al. (2001). In general, we seek the lowest order model that captures the underlying physical process and hence will generalize to other data sets. As with linear AR modeling, we create the training set on which to build our

SVM-based model by using each observation as the dependent variable and the previous p observations as independent variables. Our training samples are thus $\{(\mathbf{x}^k_{t-p:t-1}, x^k_t), t = p+1, \ldots, t_0\}$.

Ideally we would like to find a function f such that $f(\mathbf{x}^k_{t-p:t-1}) = x^k_t$ for all k and $t \leq t_0$. However, the form of f is often restricted to the class of linear functions (as is the case for AR models),

$$f(\mathbf{x}^k_{t-p:t-1}) = \langle w, \mathbf{x}^k_{t-p:t-1} \rangle \tag{3.2}$$

where $\langle , \rangle$ denotes the dot (or inner) product and w is a vector of model parameters. This restricted form makes perfect fit of the data impossible in most scenarios. As a result, we allow prediction using f to have an error bounded by ϵ, and find w under this constraint. With the recent advances in penalized regression methods such as ridge regression and lasso, the improved prediction performance of shrunken (or smoothed) models is now well-understood (Copas, 1997; Fu, 1998). Thus in order to provide a model that maximizes prediction performance, we seek to incorporate shrinkage on the model paramaters w. Such shrunken w may be found by minimizing the Euclidean norm subject to the error constraint ϵ, namely

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\|w\|^2 \\ &\text{subject to} && \begin{cases} x^k_t - \langle w, \mathbf{x}^k_{t-p:t-1} \rangle \leq \epsilon \\ \langle w, \mathbf{x}^k_{t-p:t-1} \rangle - x^k_t \leq \epsilon \end{cases} \end{aligned} \tag{3.3}$$

This model relies on the assumption that a linear model is able to fit the data to within precision ϵ. However, typically such a linear model does not exist, even for moderate settings of ϵ. As such, we introduce the slack variables ξ^+_t, ξ^-_t, to allow for deviations beyond ϵ. The resulting formulation is

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\|w\|^2 + C\sum_{t=p+1}^{t_0}(\xi^+_t + \xi^-_t) \\ &\text{subject to} && \begin{cases} x^k_t - \langle w, \mathbf{x}^k_{t-p:t-1} \rangle \leq \epsilon + \xi^+_t \\ \langle w, \mathbf{x}^k_{t-p:t-1} \rangle - x^k_t \leq \epsilon + \xi^-_t \end{cases} \end{aligned} \tag{3.4}$$

The constant C controls the tradeoff between giving small w and penalizing deviations larger than ϵ. In this form we see that only points that lie outside of the bound ϵ have an effect on w. Figure 3.1 illustrates the process graphically.

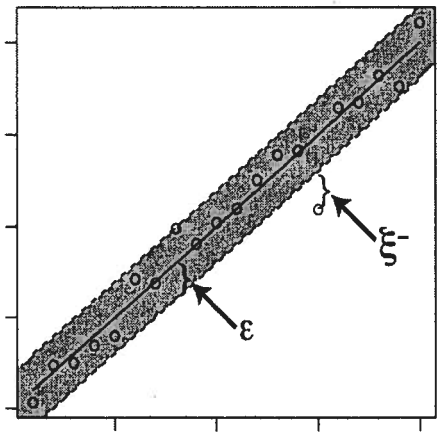


Figure 3.1: Illustration of linear support vector regression fit

Although this optimization problem is straightforward to carry out, the extension to non-linearity is revealed by the dual formulation. We thus proceed by constructing a Lagrange function of the above by introducing a set of dual variables.

$$L := \frac{1}{2}\|w\|^2 + C\sum_{t=p+1}^{t_0}(\xi_t^+ + \xi_t^-) - \sum_{t=p+1}^{t_0}\alpha_t^+\left(\epsilon + \xi_t^+ - x_t^k + \langle w, \mathbf{x}_{t-p:t-1}^k\rangle\right) \tag{3.5}$$

$$- \sum_{t=p+1}^{t_0}\alpha_t^-\left(\epsilon + \xi_t^- - x_t^k + \langle w, \mathbf{x}_{t-p:t-1}^k\rangle\right) - \sum_{t=p+1}^{t_0}\left(\eta_t^+\xi_t^+ + \eta_t^-\xi_t^-\right)$$

where the dual variables α_t^+, α_t^-, η_t^+, η_t^- are understood to be non-negative. It can be shown that this function has a saddle point at the optimal solution, and hence

$$\frac{\partial L}{\partial w} = w - \sum_{t=p+1}^{t_0}\left(\alpha_t^+ + \alpha_t^-\right)\mathbf{x}_{t-p:t-1}^k = 0 \tag{3.6}$$

$$\frac{\partial L}{\partial \xi_t^+} = C - \alpha_t^+ - \eta_t^+ = 0$$

$$\frac{\partial L}{\partial \xi_t^-} = C - \alpha_t^- - \eta_t^- = 0$$

Plugging these saddlepoint constraints into L yields the following dual

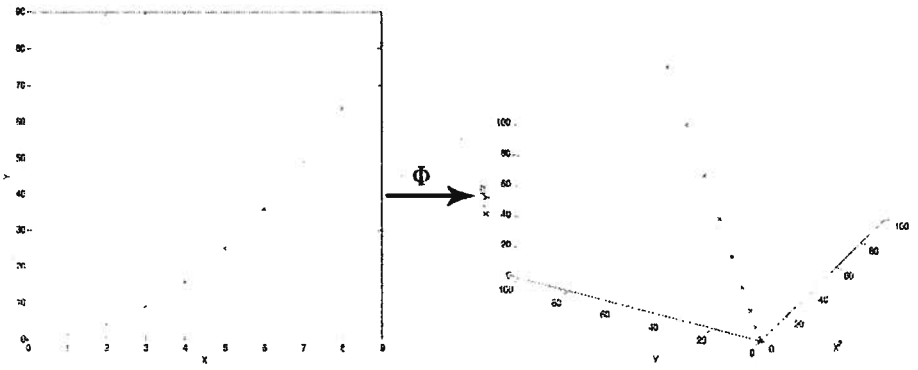


Figure 3.2: Illustration of mapping to an alternate space to induce linearity

optimization problem:

$$\begin{aligned} &\text{maximize} && \begin{cases} \frac{1}{2}\sum_{t,t'=p+1}^{t_0}\left(\alpha_t^+ + \alpha_t^-\right)\left(\alpha_{t'}^+ + \alpha_{t'}^-\right)\left\langle \mathbf{x}_{t-p:t-1}^k, \mathbf{x}_{t'-p:t'-1}^k\right\rangle \\ -\epsilon\sum_{t=p+1}^{t_0}\left(\alpha_t^+ + \alpha_t^-\right) + \sum_{t=p+1}^{t_0} x_t^k\left(\alpha_t^+ - \alpha_t^-\right) \end{cases} \\ &\text{subject to} && \begin{cases} x_t^k - \langle w, \mathbf{x}_{t-p:t-1}^k\rangle \leq \epsilon + \xi_t^+ \\ \langle w, \mathbf{x}_{t-p:t-1}^k\rangle - x_t^k \leq \epsilon + \xi_t^- \end{cases} \end{aligned} \tag{3.7}$$

Notice that by the saddlepoint constraint $w = \sum_{t=p+1}^{t_0}\left(\alpha_t^+ + \alpha_t^-\right)\mathbf{x}_{t-p:t-1}^k$ we may write f as

$$f(\mathbf{x}_{t-p:t-1}^k) = \sum_{t'=p+1}^{t_0}\left(\alpha_t^+ - \alpha_t^-\right)\left\langle \mathbf{x}_{t'-p:t'-1}^k, \mathbf{x}_{t-p:t-1}^k\right\rangle \tag{3.8}$$

In this way w may be viewed as a linear combination of the training points $\mathbf{x}_{t-p:t-1}^k$. Note also that in this formation both f and the corresponding optimization can be described in terms of dot products between the data. In this way, we can transform the data using the function $\Phi : \mathbb{R}^p \to F$, and compute the dot products in the transformed space. Such mappings allow us to extend beyond the linear framework presented above. Specifically, the mapping allows us to fit linear functions in F which, when converted back to $\mathbb{R}^p$, are nonlinear. A toy example of this process is illustrated for a mapping $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$, namely $\Phi(x, y) = (x^2, x\sqrt{(}y), y)$, in Figure 3.1. Here the data is generated using the relationship $y = x^2$. To make use of this transformed space, we replace the dot product term with

$$\left\langle \Phi\left(\mathbf{x}_{t'-p:t'-1}^k\right), \Phi\left(\mathbf{x}_{t-p:t-1}^k\right)\right\rangle \tag{3.9}$$

If F is of high dimension, then the above dot product will be extremely expensive to compute. In some cases, however, there is a corresponding kernel that is simple to compute. For example, the kernel $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d$ corresponds to a map Φ into the space spanned by all products of exactly d dimensions in $\mathbb{R}^p$. When $d, p = 2$, for instance, we have

$$\begin{aligned}(\mathbf{x} \cdot \mathbf{y})^d &= ((x_1, x_2) \cdot (y_1, y_2))^2 \\ &= \left(\left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right) \cdot \left(y_1^2, \sqrt{2}y_1y_2, y_2^2\right)\right) \\ &= (\Phi(\mathbf{x}), \Phi(\mathbf{y}))\end{aligned} \tag{3.10}$$

defining $\Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. More generally, it has been shown that every kernel that gives a positive matrix $(k(\mathbf{x}, \mathbf{y}))_{ij}$ has a corresponding map $\Phi(\mathbf{x})$ (Smola and Schlkopf, 2004). One such family of kernels we focus on is Radial Basis Function (RBF) kernels, which have the form

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2/(2\sigma^2)\right) \tag{3.11}$$

where σ^2 is the kernel variance. This parameter controls fit, with large values leading to smoother functions and small values leading to better fit. In practice moderate values are preferred as a trade-off between model fit and prediction performance.

Whereas a traditional AR(p) model employs a linear model that is a function of the previous p time points, the SVM model looks at the previous p time points compared to all groups of p successive data points from the training sample. Specifically, the model has the form

$$f(\mathbf{x}^k_{t-p:t-1}) = \sum_{j=p+1}^{t_0} \beta_j k(\mathbf{x}^k_{j-p:j-1}, \mathbf{x}^k_{t-p:t-1}) \tag{3.12}$$

Typically only a small fraction of the coefficients β_j are non-zero. The corresponding samples $\mathbf{x}^k_{j-p:j-1}$ are called support vectors of the regression function because only these select samples are used in the formulation of the model. Once we have trained our model above, we use it to predict each future observation. We then take the residuals and use them as an indicator of structural change. For our purposes we employ a control chart to monitor if the system generating the data has changed. In this discussion the control chart is created by constructing 99% control lines that correspond to 99% confidence intervals for the residuals of the model fit to the undamaged data assuming the residuals are normally distributed. This normality assumption is further discussed in the experimental results below. These control lines

are then extended through the remaining (potentially damaged) data and damage is indicated when a statistically significant number of residuals, in this case more than 1%, lie outside these lines. Note damage can also be indicated when the residuals no longer have a random distribution even though they may not lie outside the control lines.

RBF neural networks, which have the same form as Equation (3.12), have previously been used to perform SHM (e.g. Rytter and Kirkegaard (1997)). However, fitting these networks requires much more user input such as selecting which β_j are non-zero as well as selecting the corresponding training points. In addition, the fitting of the neural network model is a rather complicated nonlinear optimization process relative to the simple quadratic optimization used in the support vector framework. Although the SVM models are more easily developed, Schlkopf et al. (1997) have demonstrated that SVMs still more accurately predict the data than the RBF neural networks despite their simplicity.

3.2.1 Example: Simulated Damage

We now compare the performance of the SVM-based damage detection method to a traditional AR model with coefficients estimated by the Yule-Walker method (see, for example, Brockwell and Davis (1991)). The data is generated as follows for discrete time points, $t = 1, \ldots, 1200$:

$$x_t^1 = sin^3(400\pi t/1200) + sin^2(400\pi t/1200) + sin(200\pi t/1200) \quad (3.13)$$
$$+sin(100\pi t/1200) + \Psi + \epsilon \quad (3.14)$$

where ϵ is Gaussian random noise with mean 0 and standard deviation 0.1 and Ψ is a damage term. Three different damage cases are added to this time series at various times as defined by

$$\Psi = \begin{cases} \epsilon_1 & \text{for } t = 600, \ldots, 650 \\ \frac{1}{2}sin(1000\pi t/1200) & \text{for } t = 800, \ldots, 850 \\ \epsilon_2 & \text{for } t = 1000, \ldots, 1050 \\ 0 & \textit{otherwise} \end{cases}$$

where ϵ_1 and ϵ_2 are Gaussian random noise with mean 0 and 1, and standard deviation 0.5 and 0.2, respectively. Through the use of Ψ we attempt to simulate several different types of damaged to compare the models performance handling each. This raw signal is plotted in Figure 3.3 where it can be seen that the changes caused by the damage are somewhat subtle.

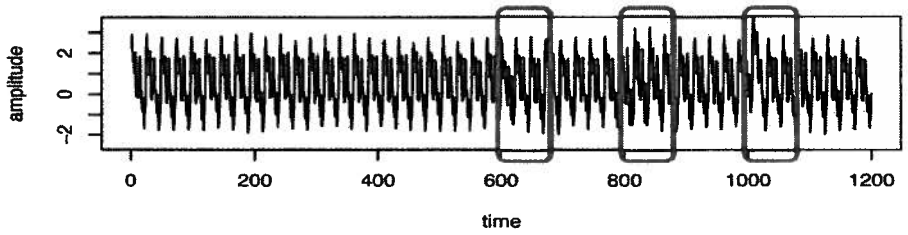


Figure 3.3: Raw simulated data with highlighted artificial damage

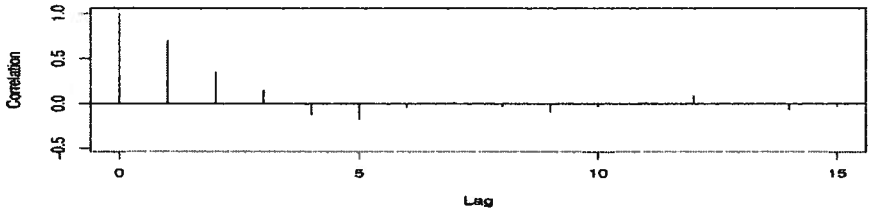


Figure 3.4: Autocorrelation plot of simulated data

The order p for both models was set at 5 as determined from the autocorrelation plot in Figure 3.4. This plot is the measure of correlation between successive time points for a given time lag. We see from the plot that after a lag of 5, the correlation is quite small, and hence little information is gained by including a longer past history p. This is a standard method for determining model order for traditional AR models, and as such should maximize this methods performance, ensuring the SVM-based model isnt afforded an unfair advantage.

The results of applying both the SVM model and a traditional AR model to the undamaged portion of the signal between time points 400 and 600 are shown in Figure 3.5 where the signals predicted by these models are overlaid on the actual signal. A qualitative visual assessment of Figure 3.5 shows that the SVM more accurately predicts this signal. A quantitative assessment is made by examining the distribution of the residual errors obtained with each model. The standard deviation of the residual errors from the SVM model is 0.26 while for the traditional AR it is 0.71, again indicating that the SVM is more accurately predicting the undamaged portion of this time series.

In order for a model to excel at detecting damage, it must fit the undamaged data well (i.e small and randomly distributed residual errors) while fitting the damaged data poorly as identified by increased residual errors

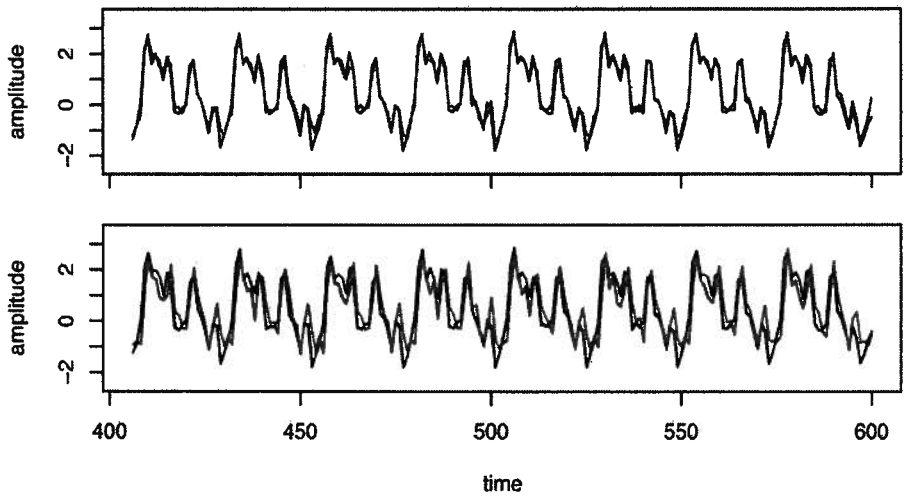


Figure 3.5: SVM (top) and linear AR models (bottom) fit to subset of data

with possibly non-random distributions. In other words, the model must be sensitive to distributional changes in the data that result from damage. To quantify such changes a control chart is developed based on the undamaged portion of the time series to establish statistically based thresholds for the damage detection process. As mentioned earlier, this control chart is calculated based on the fit to the undamaged data, specifically 99% confidence lines are drawn based on the undamaged residual error data and carried forward for comparison on the potentially damaged data. It is in this part of the process that the SVM's ability to more accurately represent the data enhances the damage detection process. The 99% confidence lines for the SVM are much closer to the mean value of the residual errors and, hence, will more readily identify small perturbations to the underlying system that produce changes in the residual error distribution. In addition, the traditional AR model shows a trend in the residuals, indicating lack of model fit, even in the undamaged case. We see that during the times of damage the residuals for the SVM-based model exceed the control limits more than occurs with the residuals from the traditional AR model. In fact, the latter method would likely miss the damage between time points 1000 and 1050, where only one point exceeds the threshold versus over 10 for the SVM-based model. This result can be seen in Figure 3.6.

Since each method performs differently for different sources of damage, it is of interest to determine when each method will be successful in indicating

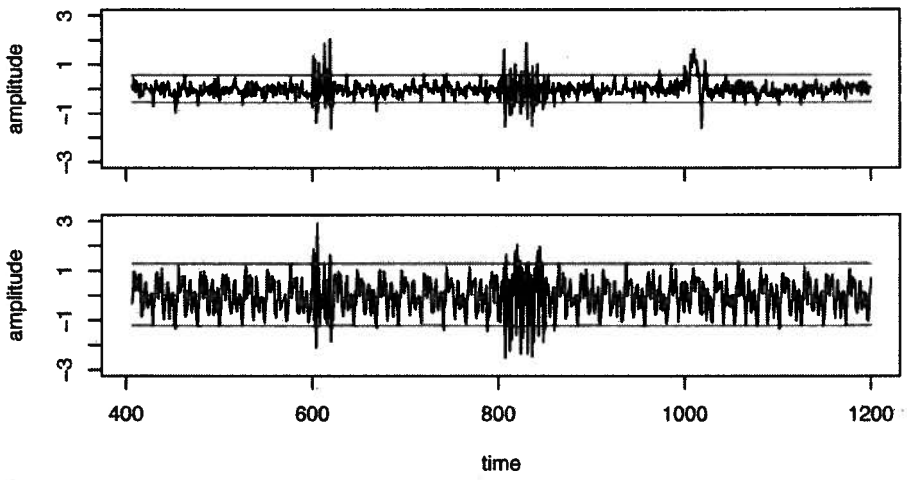


Figure 3.6: Residuals from SVM (top) and linear AR models (bottom) applied to simulated data
The 99% control lines based on the residuals from the undamaged portion of the signal are shown in red.

damage. Since the traditional AR model fits a single model to the entire data, model fit will be very poor if the data is non-stationary (for instance if the excitation is in the form of hammer impacts). Additionally, since the traditional AR model as presented above does not contain a moving average term, it will continue to fit when damage is in the form of a shift up or down in the raw time series (as demonstrated by the third damage scenario above). Conversely, the SVM-based method works by comparing each length of p data to all corresponding sets in the training set. Thus, if a similar sequence exists in the training set, we can expect the fit to be quite good. We see two scenarios in which the SVM-based method will perform poorly. Firstly, if there is some damage in the undamaged scenario, and similar damage occurs in the testing set, the model will likely fit this portion quite well. Secondly, if damage manifests itself in such a way that the time-series data is extremely similar to the undamaged time-series, the SVM methodology will be unable to detect it. However, we should emphasize that other methods, including the AR model, will suffer in such scenarios as well. As an attempted solution when the sensitivity of the method to a given type of damage is unknown and simulation tests are impossible, various damage detection methods could potentially be combined to boost detection power.

3.3 Joint Online SHM

In the undamaged state, a Gaussian distribution can often approximate the residuals from fitted models or control charts can be developed to invoke the central limit theorem and force some function of the residual errors to have a Gaussian distribution such as with an x-bar control chart (Fugate, et al, 2001). If we have K sensors, each of whose residuals are Gaussian distributed, we would like a way of combining these residuals to come up with a damage detection method that examines all K sensors. Noticing that the sum of K squared standard Gaussian random variables is distributed as a chi-squared random variable with K degrees of freedom, we square the residuals from each sensor (after they are normalized to have mean 0 and variance 1 based on the undamaged data) and add them together to create a new combined residual. These new combined residuals follow a chi-squared distribution, and hence we can make probabilistic statements about the residuals being typical or not (indicative of damage). Specifically, consider the combined residuals at some time point t:

$$\sum_{k=1}^{K}(r_t^k)^2 \tag{3.15}$$

where r_t^k is the normalized residual at time t for sensor k. Assuming the original residuals are Gaussian distributed, this random variable will have a chi-squared distribution with K degrees of freedom. Note that even when the original residuals are not approximately Gaussian, we may still employ a control chart on the combined residuals to give probabilistic statements regarding damage. For instance, when the residual errors from the fitted model have thicker tails than Gaussian, control charts must be employed to make probabilistic statements of the combined residual. However, as well see in the following example, the residual errors are often very close to Gaussian.

In addition to these combined residuals allowing us to make statements regarding damage from multiple sensor output, they also provide us with a mechanism for determining which sensors are most influenced by the damage. This latter property is of particular importance for damage location. If this combined residual is large, and hence we determine that there is damage, we can look at the values $(r_t^k)^2$ for each sensor and from their magnitudes determine which sensors contributed the most to this large combined residual. If we detect damage over a range of values, we may average $(r_t^k)^2$ over this range for each sensor to determine how much each sensor is contributing

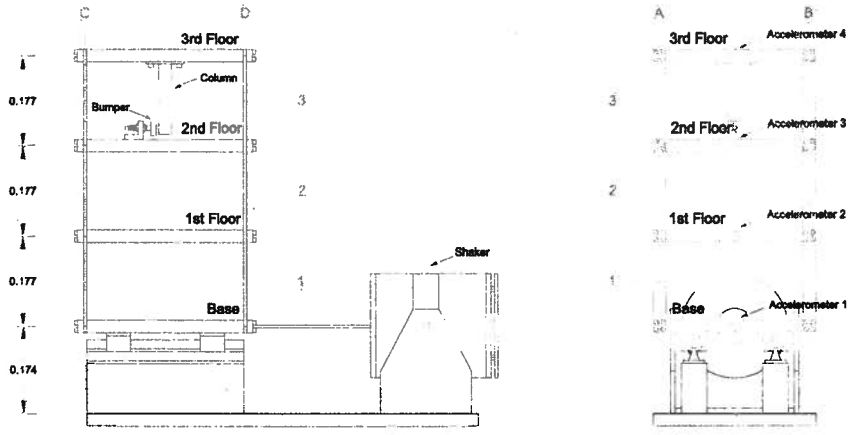


Figure 3.7: Diagram of experimental structure

to the anomalous reading.

3.3.1 Example: Experimental Data

We look at joint online SHM using SVMs on experimental data from a structure designed to produce non-linear response when it is damaged. The structure is a three-story building (Figure 3.7) consisting of aluminum columns and plates with bolted joints and a rigid base that is constrained to slide horizontally on two rails when excited by an electro-dynamic shaker. Each floor is a 30.5 × 30.5 × 2.5cm plate and is separated from adjacent floors by four 17.7×2.5×0.6cm columns. To induce non-linear behavior, a 15.0×2.5×2.5cm column is suspended from the top floor and a bumper is placed on the second floor. The contact of this suspended column with the bumper results in non-linear effects. The initial gap between the suspended column and the bumper is adjusted to simulate different levels of damage. In our test data we employ the case where the column is set 0.05mm away from the bumper. The undamaged data is obtained when the bumper and suspended column do not contact each other. The structure is subjected to a random base excitation from the shaker in both its undamaged and damaged condition. Accelerometers mounted on each floor record the response of the structure to these base excitations. A more detailed description of the test structure and the data obtained can be found at www.lanl.gov/projects/ei.

We first concatenate the undamaged data with the damaged data to demonstrate that the proposed methodology adequately detects the damage.

The SVM time series models are developed for each of the accelerometer measurements from the undamaged data as follows:

1. Select the number of time lags that will be used in the time series models. In this case eight time lags were used based on the AIC. Note the number of time lags is analogous to the order of an AR model.

2. Select the parameters of the SVM model, including the kernel type and corresponding parameters as well as C and ϵ, which control model fit as described earlier. In our case we used a Gaussian kernel with variance 1 and set $C = 1$ and $\epsilon = 0.1$. We have found the methodology to be robust to choices of variance ranging over an order of magnitude. In addition, C could be increased to force fitting of extreme values, and ϵ could be lowered to enforce a closer fit to the training data.

3. Pass the data (arranged as dependent variable and previous p points as independent variables) to the optimization described by Equation (3.7). In this case we use the first 6000 undamaged points as training data. This step is handled by the wide variety of support vector machine software available covering multiple computing environments including MATLAB and R. In particular, we employ the libSVM library with accompanying MATLAB interface (Chang and Lin, 2001).

4. Once the SVM model is trained (i.e. the β_j in Equation (3.12) are selected) in step 3, make predictions based on the new test data from the structure in its undamaged or damaged condition. Next, calculate the residual between the measured data and the output of the time series prediction.

5. Square and add the residuals from each sensor as described by Equation (3.15). Build a control chart for these combined residuals to detect damage (perhaps in conjunction with statistical tests such as a sliding window approach).

Note that steps 1 through 4 of this process are applied to each time series recorded by the four accelerometers shown in Figure 3.7.

First we will revisit the normality assumption that was made in constructing the control chart. Figure 3.8 shows the resulting Q-Q plot for the residuals from the SVM model fit to sensor 4 data obtained with the structure in its undamaged state. The Q-Q plot compares the sample quantiles of the residuals to theoretical quantiles of a Gaussian distribution. We see

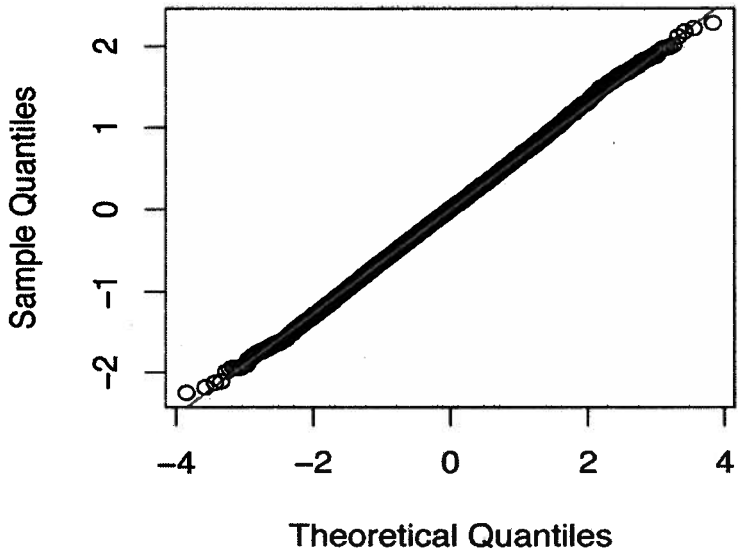


Figure 3.8: Q-Q Plot of residuals from SVM model

in this figure that the sample quantiles fall very close to the theoretical line, and hence our residuals are approximately Gaussian.

Figure 3.9 shows the residual errors from the SVM fit to each of the accelerometer readings, respectively, and the corresponding 99% control limits that are based on the first 6000 points from the undamaged portion of each signal. There are 8192 undamaged points and 8192 damaged ones. Thus when we concatenate the data the damage occurs at time point 8193 of 16384. Figure 3.10 shows the density of the normalized residual errors from all the sensors that have been combined according to Equation (3.15). We see that the distribution is very nearly chi-squared. In situations where the original residuals arent normal, this result wont be true, and hence probabilistic statements regarding the presence of damage must be made based on control charts.

Figure 3.11 shows the combined residuals as a function of time. The blue points in the plot show damage indication using the sliding window approach of Ma and Perkins (2003) as described in the introduction and based on the 99% control lines. Specifically we use a window size of 6 which, when combined with the 99% control limit, detects damage whenever 1 or more of the 6 points in the window exceeds the control line (equivalent to binomial probability of 0.05). We see from Figure 3.9 that sensors 3 and

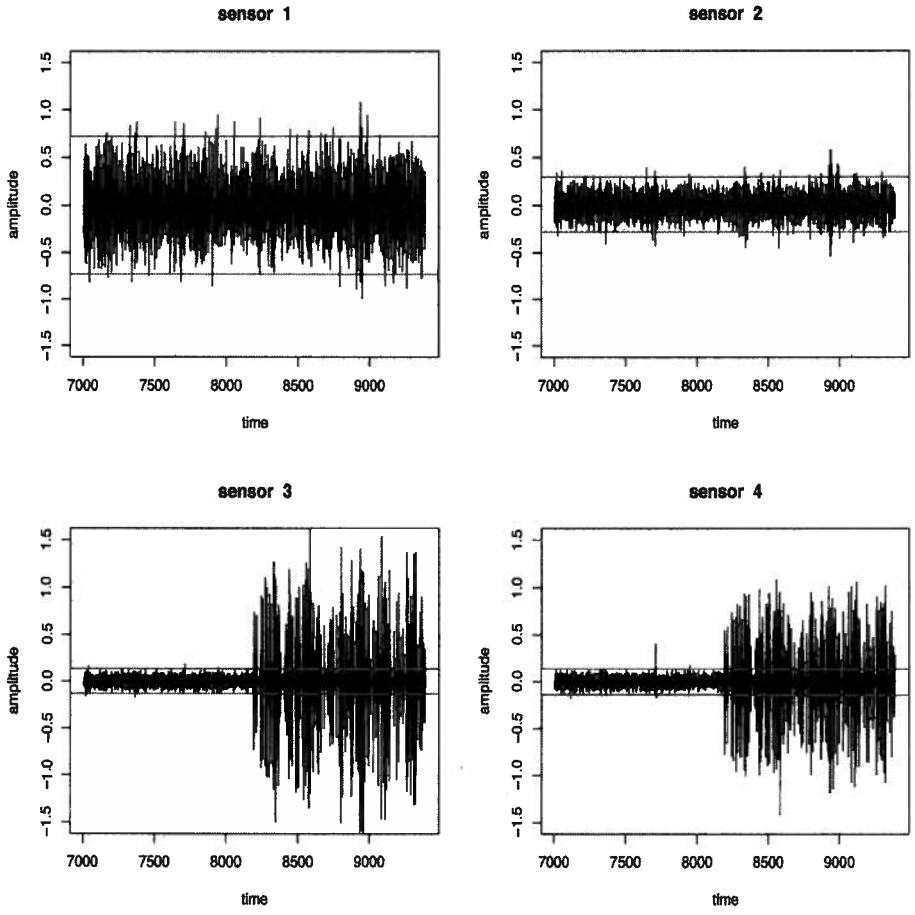


Figure 3.9: Residuals from 4 sensors for t = 7000, ... , 9384
. The 99% control lines are shown in red

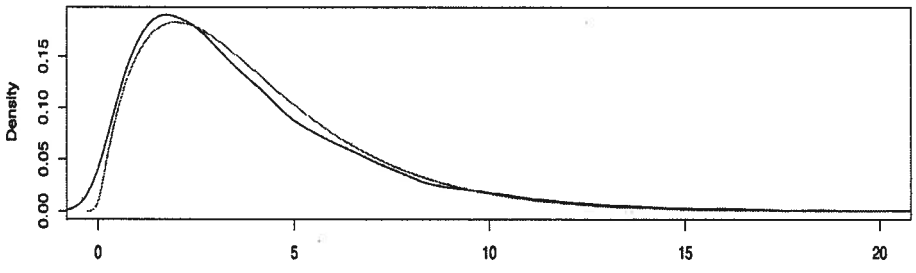


Figure 3.10: Density estimate of combined residual (black) vs. chi-squared distribution(red)

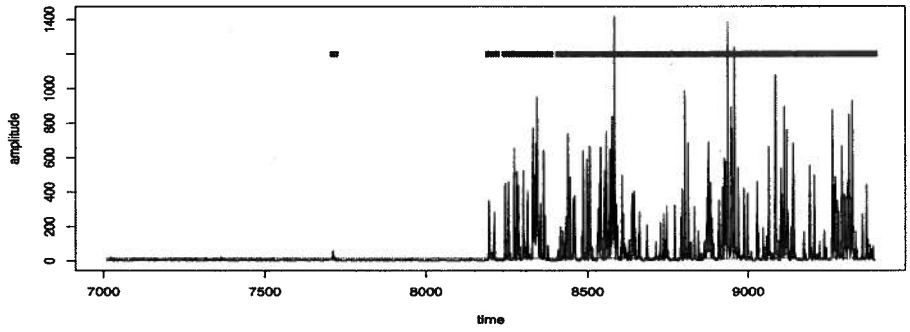


Figure 3.11: Combined residuals from all 4 sensors
The 99% control line shown in red. Sliding window damage indicator shown in blue.

4 are most influenced by damage. This result is expected as the bumper is mounted between these two sensors. In fact, if we look at the average values of $(r_t^k)^2$ (which are the individual squared residuals for sensor k) over the damaged section for each sensor, we see that the first two sensors have values 0.96 and 1.24, whereas the second two sensors have values 59.80 and 38.2, respectively. Thus from this numerical rating we can see that sensors 3 and 4 are most influenced by the damage, which agrees with the result shown in Figure 3.9.

From this analysis it is evident that we can use the combined residuals to establish the presence of damage in a statistically rigorous manner and then examine the individual sensor residuals in an effort to locate the sensors most influenced by the damage. This latter information can be used to help locate the damage assuming that the damage is confined to a discrete location such as the formation of a crack in a welded connection. Further investigation is needed to assess how this procedure could be used to locate damage for more distributed damage such as that associated with corrosion.

3.4 Conclusion

Although the application of statistical techniques to structural health monitoring has been investigated in the past, these techniques have predominantly been limited to identifying damage-sensitive features derived from linear models fit to the output from individual sensors. As such, they are typically limited to identifying only that damage has occurred. In general,

these methods are not able to identify which sensors are associated with the damage in an effort to locate the damage within the resolution of the sensor array. To improve upon this approach to damage detection, we have applied support vector machines to model sensor output time histories and have shown that such nonlinear regression models more accurately predict the time series when compared to linear autoregressive models. Here the metric for this comparison is the residual errors between the measured response data and predictions of the time series model.

The support vector machine autoregressive method is superior to traditional linear AR in both its ability to handle nonlinear dynamics as well as the structure of the model. Specifically, the support vector approach compares each new testing point to the entire training set whereas the traditional AR model finds a simple linear relationship to best describe the entire training set, which is then used on the testing data. For example, when dealing with transient impact data, the AR model will fail in trying to fit the entire time domain with a simple linear model. Whereas in the past RBF neural networks have been used to tackle this problem, these networks require significant user input and complex methods for fitting the model to the training data, and hence the simple support vector framework is preferred.

Furthermore, we have also shown how the residuals from the SVM prediction of each sensor time history may be combined in a statistically rigorous manner to provide probabilistic statements regarding the presence of damage as assessed from the amalgamation of all available sensors. In addition, this methodology allows us to pinpoint the sensors that are contributing most to the anomalous readings and therefore locate the damage within the sensor networks spatial resolution. The process was demonstrated on a test structure where damage was simulated by introducing an impact type of nonlinearity between the measured degrees of freedom. The authors acknowledge that the approach has only been demonstrated on a structure that was tested in a well-controlled laboratory setting. This approach will have to be extended to structures subjected to real-world operational and environmental variability before it can be used in practice. However, the approach has the ability to adapt to such changes through the analysis of appropriate training data that span these conditions. Therefore, follow-on studies will focus on applying this approach to systems with operational and environmental variability as well as systems that exhibit nonlinear response in their undamaged state.

3.5 References

Allen, D., Sohn, H., Worden, K., and Farrar, C. (2002). "Utilizing the Sequential Probability Ratio Test for Building Joint Monitoring." *Proc of SPIE Smart Structures Conference. San Diego, March 2002.*

Bulut, A. and Singh, A.K. and Shin, P. and Fountain, T. and Jasso, H. and Yan, L. and Elgamal, A. (2005). "Real-time Nondestructive Structural Health Monitoring Using Support Vector Machines and Wavelets." *Proc. SPIE.* 5770:180-189.

Brockwell, P., and Davis, R. (1991). *Time Series Analysis: Forecasting and Control.* Prentice-Hall.

Chattopadhyay, A., Das, S., and Coelho, CK (2007). "Damage Diagnosis Using a Kernel-based Method." *Insight-Non-Destructive Testing and Condition Monitoring.* 49:451-458.

Chang, C-J., Lin, C-J. (2001). LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Clark, G. (2008) "Cable Damage Detection Using Time Domain Reflectometry and Model-Based Algorithms." *Lawrence Livermore National Laboratory document LLNL-CONF-402567.*

Copas, J.B. (1997) "Using Regression Models for Prediction: Shrinkage and Regression to the Mean." *Statistical Methods in Medical Research.* 6:167-183.

Doebling, S., Farrar, C., Prime, M., Shevitz, D. (1998) "A Review of Damage Identification Methods that Examine Changes in Dynamic Properties." *Shock and Vibration Digest.* 30:91-105.

Farrar, C.R., Worden, K. (2007). "An Introduction to Structural Health Monitoring." *Philosophical Transactions of the Royal Society A.* 365:303-315.

Fu, W.J. (1998). "Penalized Regressions: The Bridge Versus The Lasso." *Journal of Computational and Graphical Statistics.* 7:397-416

Fugate, M., Sohn, H., and Farrar, C.R. (2001). "Vibration-Based Damage Detection Using Statistical Process Control." *Mechanical Systems and Signal Processing.* 15:707-721.

Herzog, J., Hanlin, J., Wegerich, S., Wilks, A. (2005). "High Performance Condition Monitoring of Aircraft Engines." *Proc of GT2005 ASME Turbo Expo. June 6-9, 2005.*

Ma, J., and Perkins, S. (2003). "Online Novelty Detection on Temporal Sequences." *Proc of ninth ACM SIGKDD international conference on knowledge discovery and data mining.* 613-618.

Rytter, A., and Kirkegaard, P. (1997) "Vibration Based Inspection Using

Neural Networks," *Structural Damage Assessment Using Advanced Signal Processing Procedures, Proceedings of DAMAS 97. University of Sheffield, UK.* pp. 97108.

Scholkopf, B., Sung, K.K., Burges, CJC, Girosi, F., Niyogi, P., Poggio, T. and Vapnik, V. (1997) "Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers." *IEEE Transactions on Signal Processing.* 45:2758-2765.

Schlkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., and Williamson, R. (2001). "Estimating the Support of a High-Dimensional Distribution." *Neural Computation.* 13:1443-1471.

Shimada, M. and Mita, A. and Feng, M.Q. (2006) "Damage detection of structures using support vector machines under various boundary conditions." *Proc. SPIE.* 6174:61742-61742

Smola, A.J., and Schlkopf, B. (2004). "A tutorial on support vector regression." *Statistics and Computing.* 14:199-222.

Sohn, H., Czarnecki, J., and Farrar, C.R. (2000). "Structural Health Monitoring Using Statistical Process Control." *Journal of Structural Engineering.* 126:1356-1363.

Sohn, H., Farrar, C.R., Hemez, F.M., Shunk, D.S., Stinemates, D.W., Nadler, B.R., and Czarnecki, J.J. (2004). "A Review of Structural Health Monitoring Literature from 1996-2001." *Los Alamos National Laboratory report LA-13976-MS.*

Worden, K. And Lane, A. J. (2001) "Damage Identification Using Support Vector Machines," *Smart Materials and Structures.* 10:540-547.

Worden, K. and Manson, G. (2007). "The Application of Machine Learning to Structural Health Monitoring." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Science.* 365:515-537.

Chapter 4

Conclusion

Because research in signal processing is being undertaken by physicists, computer scientists, statisticians, and engineers among others, many tools developed by one group aren't fully adopted by others. This is partially due to differences in jargon, but also because of each group's different focus and goals. However, this thesis shows that methods developed by one group for a given purpose may often be employed quite successfully by another group for an entirely different problem.

With the state of the art in particle filtering focussing on limiting degeneracy of the algorithm, it is likely that future research in the area might be applied to the material in chapter 2 to extend the scope of application. In addition, the development of support vector machines is moving toward implementing the method quickly and online, while minimizing space requirements. These advances might increase the ability of performing structural health monitoring as discussed in chapter 3 to long time series for which storage and computation becomes difficult.

While this thesis successfully implements two separate statistical methods, each is developed in a fairly specific nature, when in fact the scope of application is much more general, and may apply to problems not covered in this work. As future research, prior sensitivity and cross-validation need to be studied with the goal of easing implementation for multi-dimensional parameters. Since existing methods, including the one presented, have computational complexity which scales exponentially with dimension, alternative methods must be found. In regards to structural health monitoring, more attention must be paid to jointly modeling all sensors simultaneously, taking their correlation into effect. In addition, more studies must be undertaken to understand the effect of varying environmental conditions as well as if the initial system is slightly damaged, and hence nonlinear. Whether the solutions to these problems come from the world of signal processing is to be seen.

Both of the ideas presented in this thesis have been greeted with enthusiasm from researchers at Los Alamos National Laboratories, who daily analyze complex and computationally expensive systems. In particular, the

use of sequential Monte Carlo for prior sensitivity and cross-validation has potential to reduce the computational time of building models for understanding complex systems such as those present in biological and weapons research. In addition, the power gained from using SVM's for structural health monitoring will allow for earlier detection of damage, and hence ensure the structure's economic viability as well as the safety of operators.

Chapter 5

Technical Appendix

Proof of Theorem 1. (following along the lines of Peruggia (1997) and Epifani et al. (2005)) We seek to show that the r^{th} moment of successive importance weights is finite. So we need to find the conditions under which $\int \phi(\Theta)d\Theta$ is finite, where $\phi(\Theta) = (q(\Theta))^{1-\gamma}(q_{\setminus S}(\Theta))^{\gamma} \times (w_{\setminus S,\gamma}(\Theta))^{r}$. We expand and simplify $\phi(\Theta)$ to obtain

$$\begin{aligned}
\phi(\Theta) =& f^{1-\gamma}(\mathbf{y}|\boldsymbol{\beta},\sigma^2) \times f^{\gamma}_{\setminus S}(\mathbf{y}|\boldsymbol{\beta},\sigma^2) \times \pi(\boldsymbol{\beta}) \times \pi(\sigma^2) \times (w_{\setminus S,\gamma}(\Theta))^{r} \\
=& f(\mathbf{y}|\boldsymbol{\beta},\sigma^2) \times \left[w_{\setminus S,\gamma}(\Theta)\right]^{\gamma+r\epsilon} \times \pi(\boldsymbol{\beta}) \times \pi(\sigma^2) \\
=& (\sigma^2)^{-(\frac{n-s(\gamma+r\epsilon)}{2}-1)-1} \times \pi(\boldsymbol{\beta}) \times \pi(\sigma^2) \\
& \times \exp\{-\frac{1}{2\sigma^2}\left[(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - (\gamma+r\epsilon)(\mathbf{y}_S-\mathbf{X}_S\boldsymbol{\beta})^T(\mathbf{y}_S-\mathbf{X}_S\boldsymbol{\beta})\right]\} \\
=& \phi_1(\Theta) \times \phi_2(\Theta)
\end{aligned}$$

where

$$\begin{aligned}
\phi_1(\Theta) =& \pi(\boldsymbol{\beta}) \times \pi(\sigma^2) \times \exp\{-\frac{1}{2\sigma^2}\left[(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^T\left[\mathbf{X}^T\mathbf{X} - (\gamma+r\epsilon)\mathbf{X}_S^T\mathbf{X}_S\right](\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})\right]\} \\
\phi_2(\Theta) =& (\sigma^2)^{-(\frac{n-s(\gamma+r\epsilon)}{2}-1)-1} \\
& \times \exp\{-\frac{1}{2\sigma^2}\left[\mathbf{y}^T\mathbf{y} - (\gamma+r\epsilon)\mathbf{y}_S^T\mathbf{y}_S - \tilde{\boldsymbol{\beta}}^T\left[\mathbf{X}^T\mathbf{X} - (\gamma+r\epsilon)\mathbf{X}_S\mathbf{X}_S^T\right]\tilde{\boldsymbol{\beta}}\right]\}
\end{aligned}$$

and $\tilde{\boldsymbol{\beta}} = \left[\mathbf{X}^T\mathbf{X} - (\gamma+r\epsilon)\mathbf{X}_S\mathbf{X}_S^T\right]^{-1}\left[\mathbf{y}^T\mathbf{X} - (\gamma+r\epsilon)\mathbf{y}_S\mathbf{X}_S^T\right]$. We will show momentarily that $\left[\mathbf{X}^T\mathbf{X} - (\gamma+r\epsilon)\mathbf{X}_S\mathbf{X}_S^T\right]$ is positive definite, and hence invertible. Note that $\phi_1(\Theta)$ is proportional to a proper density for Θ when $\left[\mathbf{X}^T\mathbf{X} - (\gamma+r\epsilon)\mathbf{X}_S\mathbf{X}_S^T\right]$ is positive definite. In this case $\phi_1(\Theta)$ is upper bounded. Now $\phi_2(\Theta)$ is proportional to an inverse gamma distribution provided that both

$$\begin{aligned}
\frac{n-s(\gamma+r\epsilon)}{2} &> 1 \\
\mathbf{y}^T\mathbf{y} - (\gamma+r\epsilon)\mathbf{y}_S^T\mathbf{y}_S - \tilde{\boldsymbol{\beta}}^T\left[\mathbf{X}^T\mathbf{X} - (\gamma+r\epsilon)\mathbf{X}_S^T\mathbf{X}_S\right]\tilde{\boldsymbol{\beta}} &> 0
\end{aligned}$$

Thus, aside from showing conditions under which $\left[\mathbf{X}^T\mathbf{X} - (\gamma + r\epsilon)\mathbf{X}_S\mathbf{X}_S^T\right]$ is positive definite, we also need to find conditions guaranteeing the above 2 inequalities. We first show that $\left[\mathbf{X}^T\mathbf{X} - (\gamma + r\epsilon)\mathbf{X}_S^T\mathbf{X}_S\right]$ is positive definite. Using the Woodbury matrix identity, we see that $\left[\mathbf{X}^T\mathbf{X} - (\gamma + r\epsilon)\mathbf{X}_S^T\mathbf{X}_S\right]^{-1}$ may be written as

$$(\mathbf{X}^T\mathbf{X})^{-1} + (\mathbf{X}^T\mathbf{X})^{-1}(\gamma + r\epsilon)\mathbf{X}_S\left(\mathbf{I} - (\gamma + r\epsilon)\mathbf{X}_s^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_S\right)^{-1}\mathbf{X}_S^T(\mathbf{X}^T\mathbf{X})^{-1}.$$

Now if $\left(\mathbf{I} - (\gamma + r\epsilon)\mathbf{X}_s^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_S\right)^{-1}$ is positive definite, the second term in the above sum is positive semi-definite. This is the case when all the eigenvalues of $\mathbf{X}_s^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_S$ are less than $\frac{1}{\gamma + r\epsilon}$. $\left[\mathbf{X}^T\mathbf{X} - (\gamma + r\epsilon)\mathbf{X}_S^T\mathbf{X}_S\right]^{-1}$ may then be written as the sum of a positive definite and a positive semi-definite matrix, and hence $\left[\mathbf{X}^T\mathbf{X} - (\gamma + r\epsilon)\mathbf{X}_S^T\mathbf{X}_S\right]$ is positive definite.

Now we proceed to find conditions ensuring

$$\mathbf{y}^T\mathbf{y} - (\gamma + r\epsilon)\mathbf{y}_S^T\mathbf{y}_S - \tilde{\boldsymbol{\beta}}^T\left[\mathbf{X}^T\mathbf{X} - (\gamma + r\epsilon)\mathbf{X}_S\mathbf{X}_S^T\right]\tilde{\boldsymbol{\beta}} > 0.$$

Simple but tedious algebra gives the following expression:

$$\begin{aligned}
&\mathbf{y}^T\mathbf{y} - (\gamma + r\epsilon)\mathbf{y}_S^T\mathbf{y}_S - \tilde{\boldsymbol{\beta}}^T\left[\mathbf{X}^T\mathbf{X} - (\gamma + r\epsilon)\mathbf{X}_S\mathbf{X}_S^T\right]\tilde{\boldsymbol{\beta}} \\
=&\text{RSS} - (\gamma + r\epsilon)e_S^T\left(\mathbf{I} - (\gamma + r\epsilon)H_S\right)e_S \\
=&\text{RSS}^*_{\setminus S}(\gamma + r\epsilon)
\end{aligned}$$

which, by the theorem's conditions, is greater than 0 for argument value 1, and since $\text{RSS}^*_{\setminus S}$ is a smoothly decreasing function in its argument, it is also positive for some positive argument value less than 1. Now, we choose $\epsilon < \frac{\alpha - 1}{r - 1}$, which implies $\alpha > \gamma + r\epsilon$. By α satisfying (2.8), the conditions outlined in the proof hold. Namely, a) $\lambda_H < 1/(\gamma + r\epsilon)$, since these eigenvalues are upper bounded by 1, b) $n - s(\gamma + r\epsilon) > 2$, and c) $\text{RSS}^*_{\setminus S}(\gamma + r\epsilon) > 0$. □