

# Prediction-Based Regularization Using Data Augmented Regression

Giles Hooker, Saharon Rosset

*Department of Biological Statistics and Computational Biology, Cornell University, Ithaca, New York,*

*e-mail: giles.hooker@cornell.edu*

*School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel,*

*e-mail: saharon@post.tau.ac.il*

**Abstract:** The role of regularization is to control fitted model complexity and variance by penalizing (or constraining) models to be in an area of model space that is deemed reasonable. This is typically achieved by penalizing a parametric or non-parametric representation of the model. In this paper we advocate the use of prior knowledge or expectations about the predictions of models for regularization. This has the twofold advantage of allowing a more intuitive interpretation of penalties and priors and explicitly controlling model extrapolation into relevant regions of the feature space. This second point is especially critical in high-dimensional modeling situations, where the curse of dimensionality implies that new prediction points usually require extrapolation. We demonstrate that prediction-based regularization can, in many cases, be stochastically implemented by simply augmenting the dataset with monte-carlo data. We investigate the range of applicability of this implementation. An asymptotic analysis of the performance of Data Augmented Regression (DAR) in parametric and non-parametric linear regression, and in nearest neighbor regression, clarifies the regularizing behavior of DAR. We apply DAR to simulated and real data, and show that it is able to control the variance of extrapolation, while maintaining, and often improving, predictive accuracy.

## 1. Introduction

Given a modeling, or learning dataset  $(X, \mathbf{y}) = \{\mathbf{x}_i, y_i\}_{i=1}^n$ ,  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $y_i \in \mathbb{R}$ , the standard regularized modeling paradigm calls for fitting a model  $\hat{y} = \hat{f}(\mathbf{x})$  to this data by balancing a *loss* function  $L$ , measuring goodness of fit, and a *penalty* function  $J$ , measuring model complexity.  $l(\mathbf{y}, \hat{\mathbf{y}})$  expresses the model's success in reconstructing our learning response, and is often chosen to be a log-likelihood function. In the case that  $f(\mathbf{x}) = f(\mathbf{x}, \boldsymbol{\beta})$  is a parametric model,  $J$  is typically taken to be a norm of the parameter vector,  $J(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_q$ , with  $q \in \{0, 1, 2\}$  the most common choices ( $q = 0$  corresponding to variable selection). In non-parametric settings, the penalty function  $J$  typically measures smoothness of the model. Examples of model-based regularization approaches include ridge regression (Hoerl and Kennard, 1970):

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

the LASSO (Tibshirani, 1996):

$$\min_{\beta} \|\mathbf{y} - X\beta\|_2^2 + \lambda\|\beta\|_1$$

and combinations of them explored in Zou and Hastie (2005). The framework is also used in nonparametric regression using smoothing splines (Wahba, 1990) and total variation penalties (Mammen and de Geer, 1997). In the machine learning community, this approach has also been taken in support vector machine methods (Vapnik, 1996) and other kernel methods such as kernel logistic regression (Zhu and Hastie, 2005). A general overview of these methods is given in Bickel and Bo (2006).

In this paper we consider a different regularization approach, which starts from the fitted, or predicted, values of the model  $f(\mathbf{x})$  as the objects whose behavior we want to control by regularizing. Our motivation in taking this approach is twofold:

1. It is often easier to leverage domain knowledge to infer what a reasonable prediction should look like than what comprises likely values for the model's parameters or a smoothness measure. When we know how we want our model to predict, and extrapolate, where it has no data, we propose to formalize this knowledge into a penalty function, and apply this penalty to our models.
2. Model extrapolation is a particularly sticky point, which is not well covered by model-based regularization. There is no guarantee that controlling model complexity would control the model's extrapolation behavior. Linear models can extrapolate wildly, and even models which extrapolate as constants, like regression trees or nearest-neighbor regression, can result in high variance for prediction (see Section 3). Once we are in high-dimension (i.e.,  $d$  is large), the curse of dimensionality (Vapnik, 1996) implies that practically every new prediction point will require extrapolation from the learning data, and hence control of extrapolation behavior becomes an integral part of good prediction performance.

By way of motivation for our concern, Figure 1 provides a scatter plot of the joint distribution of three variables in the California Housing Data (Pace and Barry, 1997). We observe that the data points take up very little of the volume described in the plot. This means that non-parametric functions estimated using these variables as explanatory features will be uncontrolled over the large, empty, parts of the plot. None of the regularization methods listed above direct their efforts towards controlling behavior away from observed data points and this can lead to high variance in predictions for these regions. This distribution of explanatory variables motivated the simulation study presented in Figure 2 where we observe that even models that extrapolate as constants like regression trees exhibit very large variance in these regions.

Our general setting requires specification of a *generative distribution*  $\mu(\mathbf{x})$  for where prediction points will appear, and of a *base response model*  $p(y|\mathbf{x})$  for the responses at each  $\mathbf{x}$ . Letting  $l(f|\mathbf{x}, y)$  be the likelihood of  $f$  given  $\mathbf{x}$  and  $y$ , we

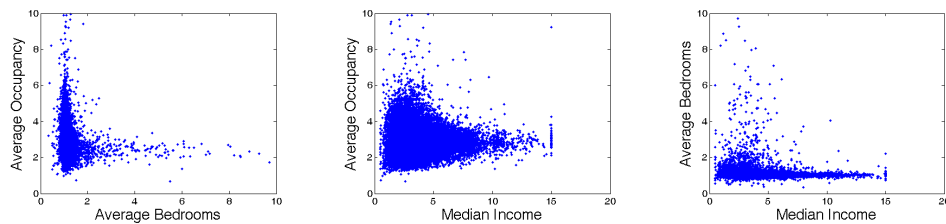


FIG 1. Scatter plots from the California Housing Data showing variables "Income", "Bedrooms" and "Occupancy".

can explicitly formulate the expected log-likelihood of prediction as a penalty function:

$$J(f|p, \mu) = \int \int l(f|\mathbf{x}, y)p(y|\mathbf{x})dp(y|\mathbf{x})d\mu(\mathbf{x}) \quad (1)$$

The regularization formulation above has the advantage that it may be applied generically. It induces a penalty over the regression function that is invariant to re-parameterization.

As we show in Section 2.1, we can sometimes solve the resulting penalized problem analytically, in one case leading to a Ridge Regression formulation, if the loss and prediction-based penalty are appropriately formed. An approach that is applicable more widely, however, is to approximate the penalty stochastically by drawing  $\mathbf{x}$  samples according to  $\mu$ , setting an appropriate response, and *augmenting* the learning sample with these additional, possibly weighted, samples. This leads to our formulation of Data Augmented Regression (DAR) as a generic algorithm. We can then also apply DAR in learning settings where no explicit loss function is being minimized, such as nearest neighbor regression (Dasarathy, 1991), by simply adding data from our prediction-penalty model to the learning sample.

In this paper we offer three main contributions:

- Proposal and formulation of prediction-based regularization algorithms, mostly implemented via DAR, covering a range of modeling problems: linear regression, generalized linear models, non-parametric regression and more (Section 2).
- Elucidation of the Bias-Variance tradeoff of DAR through analysis of its asymptotics for parametric regression, and of its finite sample behavior in the case of nearest neighbor regression (Section 4).
- Simulated and real data study of the empirical behavior of DAR (Section 3). We demonstrate its success in consistently reducing variance while maintaining or improving prediction performance.

While we are aware of several works that incorporate prior knowledge and (especially) model constraints through data augmentation or creation of virtual examples (Abu-Mostafa, 1995; Niyogi et al., 1998), we believe our work dif-

fers from these in several important respects: it treats the problem of prediction based regularization generically, rather than in the context of incorporating specific knowledge as constraints; it discusses the whole range of implementations and applications; and it offers theoretical and empirical insights into the wide applicability of this approach.

## 2. The DAR Paradigm

In this section we develop the central methodology of the paper. We begin by discussing regression in a probabilistic context in Section 2.1. We regularize the fit through a penalty, or prior, over the space of possible models, that is specified in terms of the predicted values. Doing so also induces a penalty on the parameters used to define the model. We offer a detailed analysis of this procedure for linear regression and demonstrate that under appropriate conditions it results in the familiar ridge estimates. In Section 2.2 we observe that an analytic implementation of this penalty is not always feasible and develop a stochastic implementation, which we call DAR. Section 2.3 discusses data-driven choices of hyper-parameters within the penalty. Finally, in Section 2.4 we observe that the ideas developed here do not need to be restricted to a likelihood-based optimization context and develop DAR as a general regularizing algorithm, compatible with any estimation methodology.

### 2.1. Probabilistic Prediction

Suppose that we aim to choose  $f$  from a class of models  $\mathcal{M}$  by minimizing a criterion  $\sum_{i=1}^n l(f|\mathbf{x}_i, y_i)$ , where  $\{\mathbf{x}_i, y_i\}_{i=1}^n$  is a learning sample. We add a penalty  $J(f)$  of the form (1).

We then select  $f \in \mathcal{M}$  by minimizing

$$l(f|X, y) + \lambda J(f|p, \mu). \quad (2)$$

$\lambda$  has been included here to make the regularization trade-off explicit. It typically duplicates the effect of parameters that control the variance of  $p$ . Here we show that for some natural choices of  $(\mathcal{M}, l, p, \mu)$ , the minimization of (2) translates into standard penalized regression problems.

We begin by considering a linear regression problem where  $p(y|\mathbf{x})$  is centered on zero and  $\mu(\mathbf{x})$  is uncorrelated among the features and has expectation zero. That is:

$$\begin{aligned} \mathcal{M} &= \{\mathbf{x}\boldsymbol{\beta}, \boldsymbol{\beta} \in \mathbb{R}^p\} \\ l(f|\mathbf{x}, y) &= (y - f(\mathbf{x}))^2 \\ \int yp(y|\mathbf{x})dy &= 0 \end{aligned}$$

$$\mu(\mathbf{x}) = \prod_{j=1}^d \mu_j(x_j), \quad \int x_j \mu_j(x_j) dx_j = 0$$

where an unbolded  $x_j$  indicates the  $j$ th feature variable. This yields the familiar ridge penalty (Hoerl and Kennard, 1970):

$$\begin{aligned} \lambda \int \int l(f|\mathbf{x}, y) p(y|\mathbf{x}) \mu(\mathbf{x}) &= \lambda \int \int \left( y - \sum x_j \beta_j \right)^2 p(y|\mathbf{x}) \prod \mu_j(x_j) d\mathbf{x} \\ &= \lambda \sum \int \beta_j^2 x_j^2 \mu_j(x_j) dx_j + \int \int y^2 p(y|\mathbf{x}) \prod \mu_j(x_j) d\mathbf{x} \\ &= \lambda \sum \sigma_j^2 \beta_j^2 + \sigma_y^2. \end{aligned}$$

It should be noted that the well-known formulation of ridge regression as a maximum posterior solution with Gaussian likelihood and Gaussian priors on the coefficients is a specific example of this setting, when  $\mu$  places point masses on the axes. A similar setting with an absolute loss criterion and a prior focussed on the co-ordinate axes results in the LASSO (Tibshirani, 1996).

The correspondence between ridge regression and our formulation breaks down, however, when polynomial or other nonlinear terms are added. In this case, our penalty provides a natural means by which to specify the prior covariance matrix to take into account the dependence among the predictor variables. This is shown to be effective in Section 3.

In a more general setting, consider a generalized regression context in which  $l(f|\mathbf{x}, y)$  is taken to be the log likelihood of an exponential family:

$$l(f|\mathbf{x}, y) = \sum_i T_i(y) \eta_i(f(\mathbf{x})) - A(\eta(f(\mathbf{x}))) - h(y) \quad (3)$$

with an associated penalty:

$$\lambda \int \left[ \int \left( \sum_i T_i(y) \eta_i(f(\mathbf{x})) - A(\eta(f(\mathbf{x}))) \right) p(y|\mathbf{x}) dy \right] \mu(\mathbf{x}) d\mathbf{x}.$$

Setting  $\theta_i(\mathbf{x}) = E(T_i(y)|\mathbf{x})$  and computing the inner integral gives

$$\lambda \int \left( \sum_i \theta_i(\mathbf{x}) \eta_i(f(\mathbf{x})) - A(\eta(f(\mathbf{x}))) \right) \mu(\mathbf{x}) d\mathbf{x}. \quad (4)$$

The term within brackets is a log conjugate prior for the likelihood (3). This is a natural prior distribution for the point-wise predicted values from  $f$ . This conjugate prior on predicted values can then be integrated over feature space to induce a prior over  $\mathcal{M}$  (i.e., a corresponding model-based prior), or this integral may be implemented stochastically, as discussed in the next subsection.

## 2.2. Data-Augmented Regression

Evaluating (1) analytically may not always be possible. Instead, a Monte-Carlo approximation may be employed. We approximate the penalty by:

$$\frac{\lambda}{N} \sum_{i=1}^N l(f|\tilde{\mathbf{x}}_i, \tilde{y}_i) \quad (5)$$

where  $\tilde{\mathbf{x}}_i$  is drawn randomly according to  $\mu(\mathbf{x})$  and  $\tilde{y}_i$  is generated by  $p(y|\tilde{\mathbf{x}}_i)$ . Thus, we are approximating the penalty by adding an *augmenting set*  $\{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^N$  to the *observed data*  $\{x_i, y_i\}_{i=1}^n$  and giving it weight  $\lambda/N$ . We have named this technique *Data-Augmented Regression*. As we show in Section 4, the variance associated with the Monte Carlo approximation (5) decreases at a  $1/N$  rate for parametric regression and may decrease faster in other contexts.

In some common contexts, the inner integral in (1) may be evaluated analytically, providing a reduction in variance due to Monte Carlo sampling. As noted above, regression in exponential family models is included in this context. If  $T_i(y)$  is linear in  $y$ , (4) is equivalent to using the partially-stochastic penalty

$$\frac{\lambda}{N} \sum_{i=1}^N \int l(f|\tilde{\mathbf{x}}_i, y) p(y|\tilde{\mathbf{x}}_i) dy.$$

This would require alternative algorithms for estimating  $f$  (that is, the optimization algorithm used to minimize  $l$  in training cannot simply be applied to the augmented data). However, for squared error loss, we observe that

$$\int (\tilde{y} - f(\tilde{\mathbf{x}}))^2 p(\tilde{y}|\tilde{\mathbf{x}}) dy = (m(\tilde{\mathbf{x}}) - f(\tilde{\mathbf{x}}))^2 + \sigma_y^2(\tilde{\mathbf{x}}) \quad (6)$$

where  $p(\tilde{y}|\tilde{\mathbf{x}})$  has mean and variance  $m(\tilde{\mathbf{x}}), \sigma_y^2(\tilde{\mathbf{x}})$ . The second term in the RHS is independent of  $f$  and thus DAR may be implemented in this case by using  $\{\tilde{\mathbf{x}}_i, m(\tilde{\mathbf{x}}_i)\}_{i=1}^N$  as the augmenting data set, and solving the squared error loss minimization problem on the augmented data. We can think of this as setting  $m(\tilde{\mathbf{x}})$  as a default model and the penalty as encouraging  $f(\mathbf{x})$  to provide output close to  $m(\mathbf{x})$  in the absence of observed data close to  $\mathbf{x}$ .

## 2.3. Practicalities: Choices of Priors

The analyses above have been undertaken in the context of a known prior model  $p(y|\mathbf{x})$  and feature distribution  $\mu(\mathbf{x})$ . In practice, both of these may need to be selected via data driven methods. In our analysis of ridge regression, we assumed that both the features and the response had marginal expectation of zero. In practice, it is common to pre-process the data by centering it, thus creating empirical marginal expectations of 0.

Within the context of DAR, we interpret this centering as a data-driven choice that  $E_{p(y|\mathbf{x})} \tilde{y} = \sum y_i/n$ . That is, we use a base model which is constant

across  $\tilde{\mathbf{x}}$  with the same mean as the observed response. This is useful as a generic prior model for a prediction approach as it is guaranteed to provide reasonable predictions across the feature space. There may be situations in which there are other, natural choices of prior models, which may also require fitting a low dimensional set of parameters.

In some cases, there are natural *data-dependent* choices for  $\mu(\mathbf{x})$  and  $p(y|\mathbf{x})$ , i.e., ones based on the learning sample. Some of these are related to the ideas of *transductive learning*, where in addition to the learning sample, there is a (potentially much larger) N-sample  $\{\mathbf{x}_i\}_{i=1}^N$  of data points where the response  $y$  is not observed. These may be viewed as an empirical distribution for prediction points. The idea of penalizing values at these points has been explored in Rifkin and Lippert (2007) and is beyond the scope of this paper. Where there are no obvious candidates, we propose taking  $\mu$  to be uniform on a hyper-rectangle containing the observed features. The uniform distribution is intended to ensure regularization in areas of low data density; points outside the bounding rectangle being easy to detect and flag as suspicious.

#### 2.4. A Generic Algorithm

So far, our discussion has been confined to an optimization-regularization context in which  $f$  is chosen by minimizing a criterion defined by  $l(f|\mathbf{x}, y)$  and we have proposed the addition of a penalty which may be implemented within pre-existing estimation techniques. However, the idea of regularizing by adding data to existing methods is considerably more general. There are many contexts in which the estimation of a prediction function is defined algorithmically. Here, common regularization techniques are also defined algorithmically. Examples include

- Nearest neighbors techniques (local averaging regularized by number of neighbors).
- Nadaraya-Watson and other kernel estimators (local averaging regularized by bandwidth).
- Decision and Regression trees (greedy approximation regularized by pruning).
- Neural networks (steepest descent regularized by early stopping).

For such techniques, DAR may be employed as an alternative (or complementary) form of prediction-focused regularization.

In formal terms, DAR proceeds as follows:

1. Inputs:  $\{X, \mathbf{y}\} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ , and estimation algorithm  $L : \{X, \mathbf{y}, w\} \rightarrow f(\mathbf{x})$ , where  $w$  is a vector of observation weights.
2. Set priors  $\mu(\tilde{\mathbf{x}}; X) = \mu(\tilde{\mathbf{x}}; X)$ ,  $p(\tilde{y}|\tilde{\mathbf{x}}) = p(\tilde{y}|\tilde{\mathbf{x}}; X, \mathbf{y})$ .  
**Defaults:**  $\mu(\tilde{\mathbf{x}}; X) = \prod_{j=1}^k I(\min(x_{ij}) < \tilde{\mathbf{x}}_i < \max(x_{ij}))$ ,  $p(\tilde{y}|\tilde{\mathbf{x}}; X, \mathbf{y}) = N(\tilde{y}, \text{var}(\mathbf{y}))$ .
3. Generate  $\{\tilde{X}, \tilde{\mathbf{y}}\} = \{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^N$  from  $\{\mu(\tilde{\mathbf{x}}), p(\tilde{y}|\tilde{\mathbf{x}})\}$

4. Set  $\{X^*, \mathbf{y}^*\} = \{[X, \tilde{X}], [\mathbf{y}, \tilde{\mathbf{y}}]\}$ ,  $w^* = [1_n, \lambda/N * 1_N]$  where  $[\cdot, \cdot]$  represents row-wise concatenation and  $1_n$  is a column of  $n$  ones.
5. Output:  $f = L(X^*, \mathbf{y}^*, w^*)$ .

Note that optimization based learning methods are a subset of algorithmic learners. DAR therefore represents a generic, prediction-focused regularization method across a wide class of estimators.

### 3. Experiments

Our motivation for DAR arose from the problem of extrapolation when using machine learning methods for prediction. Methods such as trees and neural networks are guaranteed to give predictions in a bounded interval. The variance of these predictions may, nonetheless, be severe in regions far from observed data. An example is given in Figure 2. For this experiment, we take features  $x_1, x_2$  to have a bivariate distribution that is uniform on the region a square of side 10 in which either  $x_1 < 1$  or  $x_2 < 1$ . Hooker (2004) observed that such "or"-structures are common in demographic data, but may be difficult to detect in high dimensions. The point  $(x_1, x_2) = (4, 4)$  lies in the convex hull of these data, but is nonetheless a point of extrapolation, because it is far from all data points.

We generated a data set of 1000 examples from this distribution giving each example a response of  $y = x_1 - x_2 + \epsilon$  with  $\epsilon$  drawn from a standard Gaussian. Prediction functions were then learned with the tree-based method CART (Breiman et al., 1984), implemented as `rpart` in the R statistical programming environment (R Development Core Team, 2007), employing the 1-SE rule for pruning, and for a neural network with 20 hidden nodes, fitted with the R package `nnet`. Both methods tend to constant values as any of the features are allowed to diverge, nominally providing stable extrapolation. Nonetheless, we observe that both may still exhibit high variability in regions of feature space without data.

The histograms in Figure 2 provide the distribution of predictions at the point  $(4, 4)$  resulting from 200 replications of this experiment. Predictions from regression trees are seen to be bi-modal; this results from their tendency to choose a first split to break one "arm" away from the other, but deciding which split to make with probability 1/2. Neural network predictions exhibit a more standard distribution, but also have a high degree of variability.

#### 3.1. A Simulated Example

We examine a simulated example in which true probability distributions are known. We take as predictor variables 500 examples of a 30 dimensional mixture of two Gaussians:

$$(x_1, \dots, x_{30}) \sim 0.5N(\mu_1, I) + 0.5N(\mu_2, I)$$

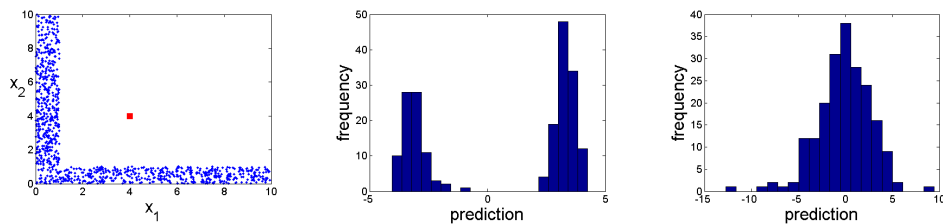


FIG 2. A simulation study of variance at extrapolation. The left hand plot provides an example of a training set on which the response is given by  $x_1 - x_2 + \epsilon$ , the square indicating the point  $(4, 4)$  at which we will evaluate trees trained on this data. The middle plot provides a histogram of CART predictions at this point for 100 samples from the data. The right hand plot gives a histogram of predictions from a 2-20-1 neural network. Both are highly variable.

in which  $\mu_1$  has value 1 in the first fifteen entries and zero in the second fifteen and  $\mu_2$  is zero in the first fifteen entries and 1 in the second. The response is given by the linear combination

$$y = \sum_{i=1}^{15} x_i - \sum_{j=16}^{30} x_j + \epsilon$$

with  $\epsilon$  iid  $N(0, 1)$ . The mixture distribution used to produce the feature variables is designed to only partially cover feature space. In particular, there is very low feature density around the line  $k * (1, \dots, 1)$  for  $k > 3$ .

In order to examine the effect of DAR we used a quadratic regression model (including all quadratic and pairwise interaction terms) and CART. We set  $\mu(x)$  to be uniform on a cube bounded by  $[-5, 6]$  in each direction. For quadratic regression, this leads to an explicit penalty matrix which has been compared with ridge regression. The DAR framework was implemented stochastically in CART using 5000 Monte-Carlo points. We evaluated the predictive performance of each estimator on data drawn from the mixture distribution, as well as examining the variance of predictions made on a data set of size 5000 drawn from  $\mu(x)$ . Results are provided in Figures 3 and 4.

For quadratic regression, it is apparent that the use of a penalty directed at predictive variance has significant advantages; it leads not only to a smaller optimal penalty, but also to a 10-fold improvement in performance for the best penalty. Moreover, the variability of predictions across  $\mu(x)$  decreases much more rapidly than for the ridge penalty. For CART, the use of an augmenting data set provides a small, but significant, improvement in performance for most weights. However, it is possible to gain a substantial decrease in predictive variance without compromising performance.

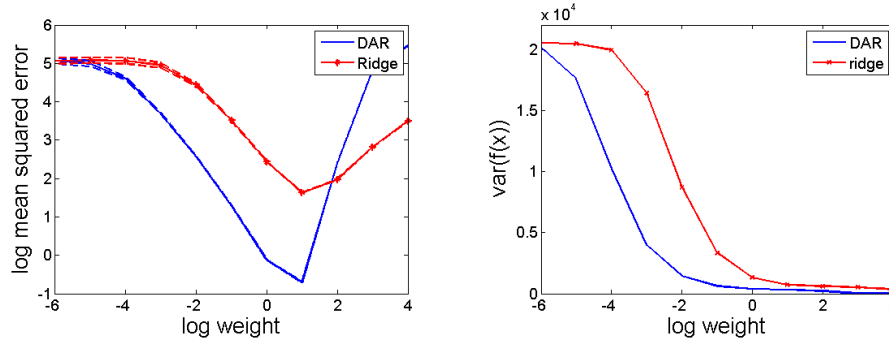


FIG 3. DAR results on a simulated data with quadratic regression. Left: log mean squared error for quadratic regression comparing DAR and ridge penalties. Right: variance of predicted values from DAR and ridge penalties on features generated from a uniform distribution.

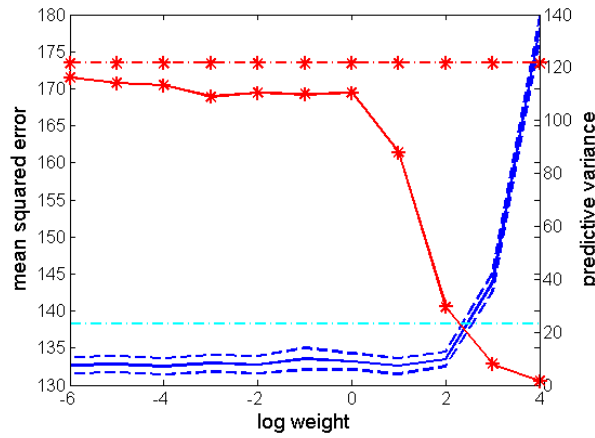


FIG 4. Results for DAR and CART on simulated data; unmarked lines indicate mean squared error with added data (solid) and without (dash-dotted). Lines marked with "\*" are the predictive variance over a uniform distribution; these are plotted with respect to the right-hand axis.

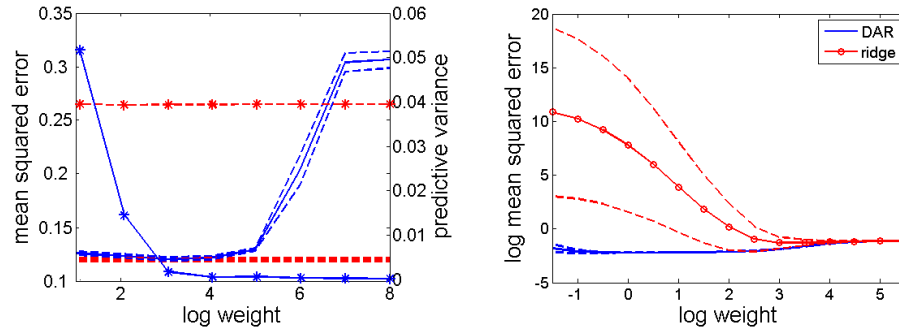


FIG 5. DAR results on the California Housing Data. Left: using CART, solid lines indicate DAR, dashed lines are without an augmenting sample. Lines marked with a '\*' represent the variance of predictions and are given with respect to the right hand axis. Right: mean squared error for quadratic regression using a DAR penalty (unmarked) and a ridge penalty (starred).

### 3.2. DAR and California Housing Data

To provide a real-world example we used the California Housing Data (Pace and Barry, 1997). This data set consists of the median house price in each of 20,460 neighborhoods of California along with nine explanatory demographic variables. For this example, we again employed CART and quadratic regression to predict the logged response. In order to assess performance on smaller training set sizes, we randomly drew 1000 samples from these data to train the model along with 5000 augmenting points drawn from a uniform distribution on the empirical range of the training data. We selected a pruning level for CART via a left-out validation set of size 100. The performance of the model was then evaluated on the remaining data. This process was repeated 100 times. This performance was compared to standard ridge regression and to CART without added data. We also evaluated the variance of predictions over these 100 simulations at each of 1000 points drawn uniformly across feature space.

Figure 5 provides the results from this analysis. It is apparent that there is significant improvement in the squared error results of quadratic regression. The variance of predictions at new points was even more significantly reduced (not shown). There is no significant improvement in the performance of CART when DAR is used. However, these performances were comparable for a large range of weight values and the variance of predictions could be substantially reduced. This fits well with our analysis of other local methods in Section 4.2. For quadratic regression, the variance of prediction for DAR was never more than 1/1000th of the corresponding variance for ridge regression, indicating a substantial improvement.

### 3.3. Further Data

Our experiments above have so far assumed that the amount of weight given to the augmenting data is known. In practise, this will also have to be estimated. In order to provide a more comprehensive study we have examined a further two data sets. These are the Boston Housing Data (Harrison and Rubinfeld, 1978), providing median housing prices in 506 suburbs of Boston along with 12 demographic variables, and the Baseball Salary Data (Christensen, 1996), which provides the salaries of 337 1991 professional baseball players along with 17 measures of their performance. In addition, we also examined the California Housing Data when data sets of size 1000 and 2000 were used for training our models.

In all cases, we examined CART and quadratic regression. In the Boston and Baseball data, we trained the model using approximately 90% of the data chosen at random. This was repeated 100 times. We selected a pruning level for CART based on a 10% validation sample, this was also used to select the amount of weight placed on the augmenting data. For quadratic and ridge regression, we estimated  $\lambda$  by the minimizing value of the generalized cross validation score (Wahba, 1990). We used an augmenting data set of size 5000 throughout. In general, we found our results to be relatively insensitive to the size of the augmenting sample. We also measured the predictions of the resulting models on 1000 uniformly-distributed points over feature space and reported the mean of the variance of these predictions over the 100 samples.

Figure 6 presents the results of these experiments. The performance CART is compared with and without an augmenting sample. DAR and quadratic regression is compared to quadratic regression with a ridge penalty. We observe that for CART, DAR in general decreased test-set performance slightly. It provided between small and significant improvement for quadratic regression. However, in both cases, there was a substantial reduction in predictive variance, indicating considerable improvement in extrapolation stability.

## 4. Added Data Asymptotics

While the DAR framework provides a direct method of incorporating prior knowledge and inducing extrapolation-resistant priors on a regression model, it may require a very large augmenting data set to do so stably and this, in turn, can produce substantial computational cost. The addition of a small amount of data, may, paradoxically, produce more variable estimates due to the variability of the augmenting set. We are therefore interested in quantifying the stability of DAR estimates in parametric and non-parametric settings. Here we offer analysis of two cases, one the standard parametric regression setting, the other a non-parametric nearest neighbor setting.

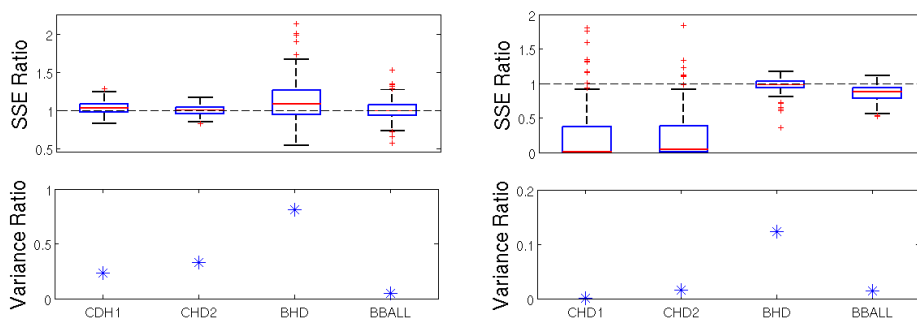


FIG 6. Performance of DAR on the California Housing Data using 1000 training observations (CHD1) and 2000 training observations (CHD2), the Boston Housing Data (BHD) and the Baseball Salary Data (BBALL). Left: CART compared with DAR and without, Right: quadratic regression with a DAR penalty versus using a ridge penalty. Top panels: ratio of squared error for DAR to squared error without DAR for 100 random samples. Bottom panels ratio of predictive variance for DAR to predictive variance without DAR over the same 100 random samples.

#### 4.1. Parametric Regression

When the regression function can be written as  $f(\mathbf{x}|\theta)$  for  $\theta \in \mathbb{R}^d$ , it is reasonable to hope that the variance of  $\hat{\theta}$  due to the implementation of a stochastic penalty will decrease with a  $1/N$  rate. Let  $l_0(\theta) = \sum_{i=1}^n l(f(\mathbf{x}|\theta)|\mathbf{x}_i, y_i)$  be an optimization criterion for  $\theta$ , then the following holds:

**Theorem 4.1.** *Suppose the augmented normal equations*

$$L(\theta) = \frac{dl_0(\theta)}{d\theta} + \lambda \int \frac{dl(\theta|\tilde{\mathbf{x}}, \tilde{y})}{d\theta} p(\tilde{y}|\tilde{\mathbf{x}}) \mu(x) d\tilde{y} d\tilde{\mathbf{x}} = 0$$

have a unique solution at  $\theta = \theta_0$  and let  $\theta_N$  uniquely satisfy the stochastically augmented version

$$L_N(\theta) = \frac{dl_0(\theta_N)}{d\theta} + \frac{\lambda}{N} \sum_{i=1}^N \frac{dl(\theta_N|\tilde{\mathbf{x}}_i, \tilde{y}_i)}{d\theta} = 0$$

where  $\{\tilde{\mathbf{x}}_i, \tilde{y}_i\}$  is independently drawn from  $\{\mu(\mathbf{x}), p(y|\mathbf{x})\}$ .

Further suppose that both  $l_0$  and  $l$  have three continuous derivatives and that their third derivatives are bounded by some  $\mu$ -integrable function  $M(\mathbf{x})$  not depending on  $\theta$ . Also suppose that the first two derivatives of  $l$  are square integrable with respect to  $\mu(\mathbf{x})$  for every  $\theta$ . Then

$$\sqrt{N}(\theta_N - \theta_0) \rightarrow N\left(0, \tilde{I}_2^{-1}(\theta_0) \tilde{I}_1(\theta_0) \tilde{I}_2^{-1}(\theta_0)\right)$$

where

$$\tilde{I}_1(\theta_0) = \lambda^2 \int \frac{dl(\theta_0|\mathbf{x}, y)}{d\theta} \frac{dl(\theta_0|\mathbf{x}, y)}{d\theta}^T p(y|\mathbf{x})\mu(\mathbf{x})dyd\mathbf{x} - \frac{dl_0(\theta_0)}{d\theta} \frac{dl_0(\theta_0)}{d\theta}^T \quad (7)$$

and

$$\tilde{I}_2(\theta_0) = \frac{d^2l_0(\theta_0)}{d\theta^2} + \lambda \int \frac{d^2l(\theta_0|\mathbf{x}, y)}{d\theta^2} p(y|\mathbf{x})\mu(\mathbf{x})dyd\mathbf{x} \quad (8)$$

The proof of this theorem is given Appendix A.

We observe that the variance due to the use of the Monte Carlo approximation may then be estimated from

$$\text{var}_{MC}(\theta_N) = \frac{1}{\sqrt{N}} \hat{I}_2^{-1}(\theta_N) \hat{I}_1(\theta_N) \tilde{I}_2^{-1}(\theta_N)$$

where  $\hat{I}_1(\theta)$  and  $\hat{I}_2(\theta)$  result from the substitution of a Monte Carlo approximation for the integrals in (7) and (8) respectively.  $\text{var}_{MC}(\theta_N)$  may then be used as a diagnostic for the size of the augmenting data set.

So far we have discussed the case that the total mass of the augmenting set is assumed to be held constant. In this situation the augmenting set does not affect the asymptotic properties of  $\hat{\theta}$  as the number of observations increases. In practice, we may want to allow both the regularization parameter  $\lambda(n)$  and the number of augmentation points  $N(n)$  to grow with  $n$ . In a likelihood framework, if  $\lambda(n)$  is sublinear, the usual MLE asymptotics still hold. For  $\lambda(n)$  superlinear, the prior dominates. For  $\lambda(n)/n \rightarrow \lambda$ , Theorem 4.1 holds if we let  $\theta_0$  solve

$$\int \frac{dl(\theta|X, y)}{d\theta} d\mu_0(X, y) + \lambda \int \frac{dl(\theta|\tilde{\mathbf{x}}, \tilde{y})}{d\theta} p(\tilde{y}|tx)\mu(\tilde{\mathbf{x}})d\tilde{y}d\tilde{\mathbf{x}} = 0.$$

and substitute

$$I_1(\theta_0) = \text{var} \frac{dl_0(\theta_0)}{d\theta} + \lambda^2 \text{var}_{p(y|\mathbf{x})\mu(\mathbf{x})} \frac{dl(\theta_0)}{d\theta} \quad (9)$$

$$I_2(\theta_0) = E \frac{d^2l_0(\theta_0)}{d\theta^2} + \lambda \int \frac{d^2l(\theta_0|\mathbf{x}, y)}{d\theta^2} p(y|\mathbf{x})\mu(\mathbf{x})dyd\mathbf{x} \quad (10)$$

When  $l_0$  represents a log likelihood, both terms that involve  $l_0$  in (9) and (10) may be replaced by the information  $I(\theta_0)$  for the original (un-augmented) parametric regression. Similarly, we can replace terms involving  $l$  by  $\tilde{I}(\theta_0)$  to provide an asymptotic variance

$$\left[ I(\theta_0) + \lambda \tilde{I}(\theta_0) \right]^{-1} \left[ I(\theta_0) + \lambda^2 \tilde{I}(\theta_0) \right] \left[ I(\theta_0) + \lambda \tilde{I}(\theta_0) \right]^{-1}$$

For procedures that cannot accept weighted data, we can only control the weight of the added data through the size of the augmenting data set as a function  $N(n)$  of the number of observed samples  $n$ . Our general experience is that in realistic situations,  $N$  may need to be large in order to obtain satisfactory

control over the variance due to the Monte Carlo approximation. Where weights are not available, we must either accept a large variance due to the augmenting data or have that data dominate the observations in fitting the data. We therefore only recommend DAR as a regularization procedure for estimation methods that accept weights.

#### 4.2. Nearest-Neighbors

Our chief area of concern is in nonparametric methods. Here we hope that DAR will stabilize prediction in regions without observed data while providing less severe regularization closer in. In this section we use a simple example to suggest that this is the case and that moreover, the variance due to the stochastic implementation of DAR is small.

Consider a nearest-neighbors prediction scheme to which DAR has been applied. Assume that the total weight of the Monte Carlo data is kept at a constant  $\lambda$ ; when  $N$  Monte-Carlo points are added each is given weight  $\lambda/N$ . For a new data point  $\mathbf{x}$ , let  $\mathbf{x}_{(1)}$  be the nearest neighbor of  $\mathbf{x}$  among the observed data,  $y_{(1)}$  its observed response and give the Monte Carlo data an ordering,  $\tilde{\mathbf{x}}_{(1)}, \tilde{\mathbf{x}}_{(2)}, \dots, \tilde{\mathbf{x}}_{(N)}$  defined by distance from  $\mathbf{x}$ , with associated Monte Carlo responses  $\tilde{y}_{(1)}, \tilde{y}_{(2)}, \dots, \tilde{y}_{(N)}$ . Define the prediction rule at  $\mathbf{x}$  by

$$\hat{y} = \begin{cases} y_{(1)} & \text{if } \|\mathbf{x} - \mathbf{x}_{(1)}\| < \|\mathbf{x} - \tilde{\mathbf{x}}_{(\lfloor N/\lambda \rfloor)}\| \\ \frac{1}{\lfloor N/\lambda \rfloor} \sum_{i=1}^{\lfloor N/\lambda \rfloor} \tilde{y}_{(i)} & \text{otherwise} \end{cases} \quad (11)$$

The following theorem provides a bound on the variance of  $\hat{y}$ . It states that there is a decay in variance at an exponential rate to either the variance of the original nearest-neighbor rule, or to the variance of the generated sample.

**Theorem 4.2.** *Let  $S_1$  be the sphere  $\{\mathbf{t} : \|\mathbf{x} - \mathbf{t}\| \leq \|\mathbf{x} - \mathbf{x}_{(1)}\|\}$  so that  $p_\mu(S_1)$  is the probability of a Monte Carlo point falling closer than  $\mathbf{x}_{(1)}$  to  $\mathbf{x}$ . Let  $\hat{y}$  be the prediction from the rule (11), then*

$$\text{var}(\hat{y}) \leq \begin{cases} \left(\frac{\lambda}{N-1}\right)^2 \nu_{S_1}^2 + \left[\text{var}(y_{(1)}) + (Ey_{(1)} - \bar{m}_{S_1})^2\right] e^{-NK_\lambda(p_\mu(S_1))} & \text{if } p_\mu(S_1) > 1/\lambda \\ \text{var}(y_{(1)}) + \left[\left(\frac{\lambda}{N-1}\right)^2 \nu_{S_1}^2 + (Ey_{(1)} - \bar{m}_{S_1})^2\right] e^{-NK_\lambda(1-p_\mu(S_1))} & \text{if } p_\mu(S_1) < 1/\lambda \end{cases}$$

where  $\nu_{S_1}^2 = \max_{\mathbf{t} \in S_1} \text{var}_{p(y|\mathbf{t})}(\tilde{y})$  and  $\bar{m}_{S_1} = E_{p(y|\mathbf{x})\mu(\mathbf{x})}(y|S_1)$  and

$$K_\lambda(p_\mu(S_1)) = \frac{1}{\lambda} \log \frac{1}{\lambda p_\mu(S_1)} + \frac{\lambda-1}{\lambda} \log \frac{\lambda-1}{\lambda(1-p_\mu(S_1))}$$

*Proof.* Set

$$P_{S_1, N, k} = \sum_{j=k}^N \binom{N}{j} p_\mu(S_1)^j (1-p_\mu(S_1))^{N-j}$$

to be the probability of at least  $k$  Monte Carlo points falling closer than  $x_{(1)}$  to  $x$ . Then, conditional on  $x_{(1)}$ , we have

$$\begin{aligned} \text{var}(\hat{y}) \leq & P_{S_1, N, \lfloor N/\lambda \rfloor} \left( \frac{\lambda}{N-1} \right)^2 \nu_{S_1}^2 + (1 - P_{S_1, N, \lfloor N/\lambda \rfloor}) \text{var}(y_{(1)}) \\ & + P_{S_1, N, \lfloor N/\lambda \rfloor} (1 - P_{S_1, N, \lfloor N/\lambda \rfloor}) (Ey_{(1)} - \bar{m}_{S_1})^2 \end{aligned} \quad (12)$$

Now suppose that  $p_\mu(S_1) > 1/\lambda$ . That is, the mass of the Monte Carlo distribution in  $S_1$  is greater than 1. Chernoff's bound gives

$$(1 - P_{S_1, N, \lfloor N/\lambda \rfloor}) \leq e^{-N \left( \frac{1}{\lambda} \log \frac{1}{\lambda p_\mu(S_1)} + \frac{\lambda-1}{\lambda} \log \frac{\lambda-1}{\lambda(1-p_\mu(S_1))} \right)}$$

For  $p_\mu(S_1) < 1/\lambda$  the converse bound can be applied on  $P_{S_1, N, \lfloor N/\lambda \rfloor}$ . Putting these together provides the result.  $\square$

The result states that there is a decay in variance at an exponential rate to either the variance of the original nearest-neighbor rule, or to the variance of the generated sample.

**Remark 1:** The first term of (12) may be made zero by replacing  $p(y|\mathbf{x})$  with  $m(\mathbf{x}) = E(y|\mathbf{x})$  as in (6). If  $m(x) = m$  is constant, this term is zero, giving an exponential decrease in variance outside a covering of data by spheres of mass  $1/\lambda$ .

**Remark 2:** This rule amounts to a local truncation of prediction: use the nearest neighbors routine inside the  $1/\lambda$  covering of observed data and revert to  $m(x)$  otherwise. This behavior leads to no change in performance for points within the  $1/\lambda$  covering, but will stabilize points outside. This behavior is in evidence in the experiments in Section 3, in which regression trees provide only small improvement in predictive accuracy, but do show a large decrease in the variance of predictions for points drawn from  $\mu(x)$ .

**Remark 3:** A more realistic prediction rule may be to define  $k = \min\{j : \sum_{i=1}^j w_{(i)} > 1\}$  for some  $C$  and set

$$\hat{y} = \frac{\sum w_{(i)} y_{(i)}}{\sum w_{(i)}}.$$

For the DAR framework we take  $w_i = 1$  on the observed data and  $w_i = \lambda/N$  on the augmenting set. Under this scenario, we obtain the same convergence rate as Theorem 4.2 when  $p_\mu(S_1) < 1/\lambda$ . For the reverse situation, the variance is asymptotically  $(1 + \lambda p_\mu(S_1))^{-1} \sigma_\epsilon^2$  leading to a regularization of the standard nearest-neighbor regression.

In expectation, this prediction rule leads to a kernel which decays linearly in  $\|\mathbf{x}_{(1)} - \mathbf{x}\|$  with rate  $\lambda$ .

**Remark 4:** The quantities in (12) that depend on the Monte-Carlo data are calculable in a finite-data setting. For  $\mu$  uniform and  $m$  constant, denote by  $\Omega$

the support of  $\mu$ , and assume that  $\mathbf{x}$  is closer to  $\mathbf{x}_{(1)}$  than the boundary of  $\Omega$ , so that  $S_1 \subset \Omega$ . Then

$$p_\mu(S_1) = \frac{(2\pi)^{d/2} \|\mathbf{x} - \mathbf{x}_{(1)}\|^{d/2}}{V\Gamma((n+1)/2)}$$

for  $V$  the volume of  $\Omega$ . Assume  $p(y|\mathbf{x})$  to place point mass at some constant  $m$ . The variance of  $\hat{y}$  at  $\mathbf{x}$ , conditional on the observed responses, is

$$\text{var}_{MC}(\mathbf{x}) = P_{S_1, N, \lfloor N/\lambda \rfloor} (1 - P_{S_1, N, \lfloor N/\lambda \rfloor}) (y_{(1)} - m)^2$$

which can be calculated exactly. We now need to average  $\text{var}_{MC}(\mathbf{x})$  over  $\Omega$ . To do this, we average over either the observed points:

$$\text{var}_{MC}^o = \frac{1}{n} \sum_{i=1}^n \text{var}_{MC}(\mathbf{x}_i)$$

or the augmenting points:

$$\text{var}_{MC}^a = \frac{1}{N} \sum_{i=1}^N \text{var}_{MC}(\tilde{\mathbf{x}}_i).$$

The equivalent calculation may be made for metrics other than squared error and can be used to develop heuristics for how many augmenting points should be used. However, these will not substitute for direct verification with a particular learning method.

## 5. Conclusion

This paper develops a generic framework that replaces regularization and Bayesian methods that are focussed on *parameters* with methods that treat *predictions*. Frequently, we have a better prior understanding of the values we are likely to see in future data than of parameters in potentially large and complex models. Moreover, by focussing on predictive regularization, we are able to explicitly control the extrapolatory behavior of the models we fit. The proposed framework fits naturally into a statistical conceptualization of priors and regularization when likelihood-based regression methods are used. However, the framework extends directly to methods that do not use likelihoods, such as the methods of the type Breiman (2001) refers to as "algorithmic modeling".

We have produced analyses of both parametric and algorithmic models that shed light on the statistical properties of these methods and provide tools for empirically evaluating the stability of the proposed estimates. We have demonstrated the advantages of this framework on real and simulated data for both parametric and algorithmic regression. Many interesting questions remain open as areas of future research. In particular, we have not discussed the relationship of DAR to transductive learning; our initial experiments have failed to find much difference between them. Additional questions involve the application of empirical and hierarchical Bayes methods for selecting prior parameters.

## References

- Abu-Mostafa, Y. (1995). Hints. *Neural Computation* 7, 639 – 671.
- Bickel, P. and L. Bo (2006). Regularization in statistics. *Test*, 271 – 344.
- Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*.
- Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*. Wadsworth.
- Christensen, R. (1996). *Analysis of Variance, Design and Regression: Applied Statistical Methods*. New York: Chapman and Hall.
- Dasarathy, B. (1991). *Nearest Neighbor Pattern Classification Techniques*. IEEE Computer Society Press.
- Harrison, D. and D. L. Rubinfeld (1978). Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management* 5, 81–102.
- Hoerl, A. and R. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(3), 55 – 67.
- Hooker, G. (2004). Diagnosing extrapolation: Tree-based density estimation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Lehmann, E. L. and G. Casella (1998). *Theory of Point Estimation*. New York: Springer.
- Mammen, E. and S. V. de Geer (1997). Locally adaptive regression splines. *Annals of Statistics* 25(1), 387 – 413.
- Niyogi, P., F. Girosi, and T. Poggio (1998). Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*. 86(11), 2196–2209.
- Pace, R. K. and R. Barry (1997). Sparse spatial autoregressions. *Statistics and Probability Letters* 33, 291–297.
- R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.
- Rifkin, R. M. and R. A. Lippert (2007). Value regularization and the fenchel duality. *Journal of Machine Learning research*.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B* 58(1), 267 – 288.
- Vapnik, V. (1996). *The Nature of Statistical Learning Theory*. Springer-Verlag.
- Wahba, G. (1990). *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics.
- Zhu, J. and T. Hastie (2005). Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics* 14, 185 – 205.
- Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society* 67, 301–320.

### Appendix A: Proof of Theorem 4.1

We take a Taylor expansion of  $L_n(\theta)$  around  $\theta_0$  to give us that for some  $\theta^*$  on the line segment between  $\theta_0$  and  $\theta_N$ ,

$$L_{jN}(\theta_0) + (\theta_N - \theta_0)^T \frac{dL_{jN}(\theta_0)}{d\theta} + \frac{1}{2}(\theta_N - \theta_0)^T \frac{d^2L_{jN}(\theta^*)}{d\theta^2}(\theta_N - \theta_0) = 0$$

for each derivative  $j \in 1, \dots, d$ .

Re-arranging gives

$$\sqrt{N}(\theta_N - \theta_0)^T \left[ \frac{dL_{jN}(\theta_0)}{d\theta} + \frac{1}{2} \frac{d^2L_{jN}(\theta^*)}{d\theta^2}(\theta_N - \theta_0) \right] = -\sqrt{N}L_{jN}(\theta_0). \quad (13)$$

We now examine each of the terms in (13). Firstly, taking  $L_N(\theta_0)$  as a vector of the  $L_{jN}(\theta_0)$

$$\begin{aligned} \sqrt{N}L_N(\theta_0) &= \sqrt{N} \frac{1}{N} \sum_{i=1}^N \left( \frac{dl_0(\theta_0)}{d\theta} + \lambda \frac{dl(\theta_0|\tilde{\mathbf{x}}_i, \tilde{y}_i)}{d\theta} \right) \\ &\xrightarrow{d} N(0, \tilde{I}_1(\theta_0)) \end{aligned}$$

using the definition of  $\theta_0$  and the central limit theorem.

Next, we have

$$\begin{aligned} \frac{dL_N(\theta_0)}{d\theta} &= \frac{d^2l_0(\theta_0)}{d\theta_0^2} + \frac{\lambda}{N} \sum_{i=1}^N \frac{d^2l(\theta_0|\tilde{\mathbf{x}}_i, ty_i)}{d\theta^2} \\ &\xrightarrow{P} \frac{d^2l_0(\theta_0)}{d\theta^2} + \lambda \int \frac{d^2l(\theta_0|\mathbf{x}, y)}{d\theta^2} p(y|\mathbf{x}) \mu(\mathbf{x}) dy d\mathbf{x} \end{aligned}$$

by the strong law of large numbers.

Finally,

$$\begin{aligned} \left\| \frac{d^2L_{jN}(\theta_0)}{d\theta^2}(\theta_N - \theta_0) \right\| &\leq \frac{d(1+\lambda)}{n} \sum_{i=1}^N \|M(\tilde{\mathbf{x}}_i, \tilde{y}_i)\| \|\theta_N - \theta_0\| \\ &\xrightarrow{P} 0 \end{aligned}$$

where we have used that  $\theta_N \xrightarrow{P} \theta_0$ . This is shown in Lemma A.1. Putting these results together yields the theorem.

**Lemma A.1.** For  $\theta_0, \theta_n$  defined in Theorem 4.1,

$$\theta_N \xrightarrow{P} \theta_0$$

The proof of this Lemma is exactly that of part (a) of Theorem 5.1 in Lehmann and Casella (1998, pp. 463-465) with the likelihood function replaced by

$$l_N(\theta) = \sum_{i=1}^N (l_0(\theta) + l(\theta|\tilde{\mathbf{x}}_i, \tilde{y}_i))$$

and will be omitted here. We note that the assumption of uniqueness for  $\theta_N$  simplify the statement of Theorem 4.1 somewhat, but standard techniques (see Lehmann and Casella, 1998) can be used to allow for the case where there are multiple local minima.