Ridge Regression Confidence Machine

Ilia Nouretdinov I.NOURETDINOV@CS.RHUL.AC.UK
Tom Melluish T.MELLUISH@CS.RHUL.AC.UK
Volodya Vovk V.VOVK@CS.RHUL.AC.UK
Computer Science, Royal Holloway University of London, Egham, Surrey, TW20 0EX, United Kingdom

Abstract

Recently kernel based methods for machine learning have shown good performance on a range of standard problems in areas such as pattern recognition, density estimation and regression estimation. One attractive feature of these methods is that they usually make very few assumptions about the data's underlying distribution; typically requiring no more than that the data should be independently identically distributed (iid). A weakness of these methods has been that they give 'bare' predictions, i.e., they predict labels with no associated confidence information. In risk-sensitive settings such as medical diagnosis confidence information is highly desirable. Several approaches have been taken to obtain confidence information. PAC analysis can provide bounds on the probabilty of error but these can return values larger than 1. Bayesian approaches can give strong confidence bounds but require one to make a priori assumptions about the data's distribution and if these assumptions do not reflect the true distribution the bounds will not be valid. We give an algorithm which allows reasonably tight tolerance intervals for regression estimates to be found without assuming anything above iid.

1. Main idea

This paper describes a method of finding confidence information about predictions. Why is this desirable? Consider two investors, A and B who are each considering investing in two companies making their IPOs, C and D. They are both interested in how profitable the companies are likely to be after the first year. A uses regression estimation and predicts figures of 150 and 200 respectively. B uses regression estimation augmented with confidence measures and obtains 90% tolerance regions 140–160 and 50–350. Investor B is now in a much better position to make an informed purchase than investor A.

As we mentioned in the abstract, our approach exploits an underlying assumption of many machine learning methods, including the theory of Ridge Regression, that the data in training and test sets are independently identically distributed but makes no further assumptions. More specifically, we assume that some training set $(z_1, \dots, z_l) = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$ and a new example $\mathbf{x}_{l+1}$ are given; the label y_{l+1} of the new example is withheld. Our only assumption is that the sequence $(z_1, \dots, z_{l+1}) = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{l+1}, y_{l+1}))$ is *iid*; in other words, that all $(\mathbf{x}_i, y_i)$, $i = 1, \dots, l+1$, are generated independently from the same probability distribution. We wish to state something about our confidence in different values $\hat{y}$ for the label y_{l+1} of the new example.

The idea of our approach is to look at all possible completions

$$(z_1, \dots, z_{l+1}) = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{l+1}, \hat{y})) \qquad (1)$$

of the known data set $((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), \mathbf{x}_{l+1})$ and for every possible completion (1) to estimate how typical it is under the iid assumption. If completion (1) is untypical, we exclude the possibility that $y_{l+1} = \hat{y}$, and we report as our prediction for y_{l+1} the set of $\hat{y}$ for which (1) is typical. This is interval prediction, as opposed to the more conventional point predictions (although the name 'interval prediction' should not be understood literally here, since the prediction can be a region different from an interval). In statistics such prediction regions are known as tolerance regions.

In the next section we will discuss how to estimate the typicalness of a sequence such as (1). The material of this and the next sections is standard (see, e.g., (Saunders et al., 1999; Vovk et al., 1999; Vovk & Gammerman, 2000)). Starting from Section 3, we will describe this paper's main contribution: an efficient algorithm for computing tolerance regions in the case

of ridge regression.

2. P-values

In this section we will discuss ways to estimate the typicalness of a sequence $z_1, \dots, z_n$ of elements of some sample space $\mathcal{Z}$ (in typical applications, $n = l + 1$ and $\mathcal{Z}$ is the Cartesian product $\mathcal{X} \times \mathcal{Y}$ of the space $\mathcal{X}$ from which unlabelled examples $\mathbf{x}_i$ are taken and the space $\mathcal{Y}$ from which labels y_i are taken; in the case of regression, $\mathcal{Y} = \mathbb{R}$). Formally, the typicalness of a sequence $z_1, \dots, z_n$ will be measured by a *p-value function* $t : \mathcal{Z}^n \to [0, 1]$, which is required to satisfy the condition

$$P^n\{(z_1, \dots, z_n) : t(z_1, \dots, z_n) \le r\} \le r, \qquad (2)$$

for any $r \in [0, 1]$ and for any probability distribution P in $\mathcal{Z}$. If the value $t(z_1, \dots, z_n)$ is small, this suggests that the sequence $z_1, \dots, z_n$ is untypical, and (2) ensures the validity of this interpretation: for example, the probability (under any iid distribution) that $t(z_1, \dots, z_n) \le 1\%$ never exceeds 1%; therefore, this event is unlikely under the iid model. This property allows us to use the function to exclude sequences which are 'impossible' at some confidence level, e.g. 1%. Notice that this argument is valid only if the p-value function is chosen before the data $z_1, \dots, z_n$ is disclosed; we will always assume this.

To summarize, we measure the typicalness of a sequence $z_1, \dots, z_n$ by the value $t(z_1, \dots, z_n)$ taken by a p-value function t (which should be chosen in advance). The latter is required to satisfy (2). The name 'p-value function' is chosen because of the similarity of this notion to p-values in traditional statistics.

The most useful for us way of obtaining p-value functions is as follows. With every example z_i we associate some *strangeness value* α_i, and define a p-value function t by

$$t(z_1, \dots, z_n) = \frac{\#\{i = 1, \dots, l+1 : \alpha_i \ge \alpha_{l+1}\}}{l+1}. \qquad (3)$$

We will discuss the conditions under which this definition satisfies (2) in a moment, but first we explain the intuition behind this choice of t. If we have an iid sequence of examples $z_1, z_2, \dots, z_n$ then the distribution over all permutations of the sequence is uniform; this is our 'null hypothesis' that we would like to test. The alternative hypothesis is that the last element z_n is not generated by the same distribution as the rest of the sequence (remember that, in our application, z_n is the pair $(\mathbf{x}_{l+1}, \hat{y})$ in (1), which is *not* generated by the iid model unless $\hat{y} = y_{l+1}$). Therefore, it is natural to pay a special attention to the strangeness of the last element.

It can be easily shown (Saunders, 2000) that the validity condition (2) is satisfied if the strangeness values are produced by a function F in the following way:

$$\alpha_i = F(\{z_1, \dots, z_n\}, z_i), \quad i \in 1, \dots, n; \qquad (4)$$

it is important that F receives as an input the *unordered* set $\{z_1, \dots, z_n\}$ (alternatively, we can assume that F is fed with the sequence $z_1, \dots, z_l$ but the function should have the same value no matter what the ordering is). Indeed, if α_i are computed using (4), the uniform distribution over all permutations of the sequence $z_1, \dots, z_n$ implies the uniform distribution over all permutations of the sequence $\alpha_1, \dots, \alpha_n$, which, in its turn, implies that (3) satisfies (2). For example, we can say that the probability of the last member of the sequence (the new example) being the strangest is $\frac{1}{l+1}$ (provided all alphas are different with probability 1).

3. Confidence for regression estimation

Now we return to our special case: we have a training set $(z_1, \dots, z_l) = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$ and a new example $\mathbf{x}_{l+1}$. Suppose we are interested in some particular guess $\hat{y}$ for y_{l+1}; set $z_{l+1} = (\mathbf{x}_{l+1}, \hat{y})$. To find strangeness values $\alpha_1, \dots, \alpha_{l+1}$, we construct a function $f : \mathcal{X} \to \mathbb{R}$ which approximates the data $z_1, \dots, z_{l+1}$ and does not depend on the order in which $z_1, \dots, z_{l+1}$ are given; we then find the residuals $\{\alpha_i = |y_i - f(\mathbf{x}_i)|, i = 1, \dots, l+1\}$ and use them as our strangeness values. Since this is a special case of construction (4), we can define the corresponding p-value function (2); choosing some threshold (or *significance level*) r (such as 1%) we can define our predictive region $R(\mathbf{x}_1, y_1, \dots, \mathbf{x}_l, y_l, \mathbf{x}_{l+1})$ as the set of all $\hat{y}$ for which $t(z_1, \dots, z_{l+1}) > r$. This predictive region is a valid $100(1-r)\%$ *tolerance region*, in the sense of the following simple theorem.

Theorem 1 *The function R satisfies*

$$P^{l+1}\{\ (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{l+1}, y_{l+1}) : y_{l+1} \in R(\mathbf{x}_1, y_1, \dots, \mathbf{x}_l, y_l, \mathbf{x}_{l+1})\ \} \ge 1 - r \qquad (5)$$

for any probability distribution P on the possible example/label pairs.

This theorem shows that the predictions output by R have a clear probabilistic meaning; say, a 99% tolerance region will cover y_{l+1} with probability at least 99%. Note that the theorem requires randomisation over the training sequences and $\mathbf{x}_{l+1}$ as well as y_{l+1}.

Proof:

Recall that the strangeness values α_i are defined as:

$$\alpha_i = |y_i - f(\mathbf{x}_i)|, i = 1, \ldots, l.$$

Let us fix some $k = 1, \ldots, l+1$ and consider the following sets for $i = 1, \ldots, l+1$:

$$U_{k,i} = \{(\alpha_1, \ldots, \alpha_{l+1}) \quad : \quad \text{at least } k+1 \text{ of alphas are } \geq \alpha_i \text{ and at least } l-k+1 \text{ of alphas are } \leq \alpha_i\}$$

As alphas are generated independently by the same mechanism, $U_{k,i}$ has the same probability for each i; also, their union contain all possible sequences; consequently, $P^{l+1}(U_{k,i}) \geq \frac{1}{l+1}$.

Now let us fix $i = l+1$ and consider all the sets $U_k = U_{k,l+1}$. Again, their union is the set of all sequences and each has probability $\geq \frac{1}{l+1}$. It follows from this that for any $r = \frac{k+\epsilon}{l+1}, k = 1, \ldots, l, 0 \leq \epsilon < 1$:

$$\begin{aligned} & P^{l+1}\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{l+1}, y_{l+1}) : \\ & \qquad y_{l+1} \in R(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_l, y_l, \mathbf{x}_{l+1})\} \\ = \; & P^{l+1}\{\#\{i : \alpha_i \geq \alpha_{l+1}\} > r(l+1)\} \\ = \; & P^{l+1}\{\#\{i : \alpha_i \geq \alpha_{l+1}\} > k\} \\ = \; & P^{l+1}(\cup_{k+1 \leq j \leq l+1} U_j) \\ = \; & \frac{l-k+1}{l+1} = 1 - \frac{k}{l+1} \geq 1 - r \end{aligned}$$

□

However, it is clear that the above construction of R does not solve the practical problem of prediction with confidence: it is impossible to explicitly consider all possible values $\hat{y}$ for the label y_{l+1}. The main contribution of this paper is an efficient way of computing R (see Algorithm 1 below).

4. Ridge regression

Suppose $\mathcal{X} = \mathbb{R}^d$ (examples are vectors consisting of n attributes). To approximate the data, the well known ridge regression procedure (Hoerl & Kennard, 1970; Saunders et al., 1998) recommends calculating the value $\mathbf{w} = \mathbf{w}_0$ where

$$a\|\mathbf{w}\|^2 + \sum_{i=1}^{l+1} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 \to \min$$

is attained; a is a positive constant. The ridge regression approximation is then $f(\mathbf{x}) = \mathbf{w}_0 \cdot \mathbf{x}$.

We can naturally represent the method in matrix form. Let Y be the vector $(y_1, \ldots, y_l, \hat{y})'$ and X be the matrix formed from the training examples

$$X = (\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_l \mathbf{x}_{l+1})' \tag{6}$$

Now we can represent the ridge regression procedure as

$$a\|\mathbf{w}\|^2 + \|Y - X\mathbf{w}\|^2 \to \min$$

or

$$Y'Y - 2\mathbf{w}'X'Y + \mathbf{w}'(X'X + aI)\mathbf{w} \to \min$$

Taking the derivative in $\mathbf{w}$ we obtain

$$2(X'X + aI)\mathbf{w} - 2X'Y = 0$$

or

$$\mathbf{w} = (X'X + aI)^{-1}X'Y$$

So the vector $\Delta = |Y - X\mathbf{w}|$ of residuals $\alpha_1, \ldots, \alpha_{l+1}$ can be written in the form

$$\begin{aligned} \Delta &= |Y - X(X'X + aI)^{-1}X'Y| \\ &= |(I - X(X'X + aI)^{-1}X')Y| \end{aligned}$$

Note that $Y = (y_1, \ldots, y_l, 0)' + (0, \ldots, 0, \hat{y})'$ and so Δ can be represented as $|A + B\hat{y}|$ where

$$A = (I - X(X'X + aI)^{-1}X')(y_1, \ldots, y_l, 0)'$$

and

$$B = (I - X(X'X + aI)^{-1}X')(0, \ldots, 0, 1)'$$

Therefore each α_i varies piecewise-linearly as we change $\hat{y}$.

We see that our p-value p for $\hat{y}$ can only change at points where $\alpha_i = \alpha_{l+1}$ for some $i = 1, \ldots, l$. This means that we can calculate which points on the real line will have a p-value exceeding r rather than having to try all possible values for y_{l+1}, leading us to our efficient algorithm.

For each training example define

$$S_i = \{\hat{y} : \alpha_i \geq \alpha_{l+1}\} = \{\hat{y} : |a_i + b_i\hat{y}| \geq |a_{l+1} + b_{l+1}\hat{y}|\}$$

(the set of all possible labels for $\mathbf{x}_{l+1}$ such that z_i's residual is greater than or equal to z_{l+1}'s residual), where $i = 1, \ldots, l+1$. Each set S_i (always closed) will either be an interval (Fig. 1.a), a ray (Fig. 1.c), the union of two rays (Fig. 1.b), the real line (Fig. 1.d, 1.e) or empty (Fig. 1.f); it is also possible for S_i to be a point (Fig. 1.d with α_i and α_{l+1} swapped around), but this is a special case of an interval.

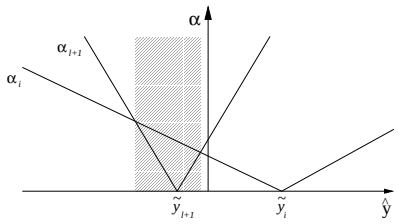

Figure a. $[u_i, v_i]$

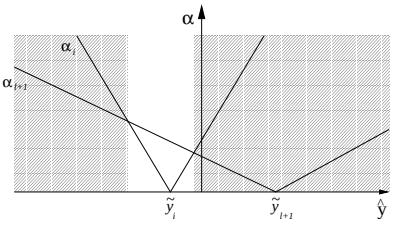

Figure b. $(-\infty, u_i] \cup [v_i; \infty)$

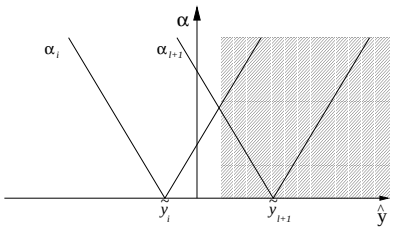

Figure c. $[u_i, \infty)$

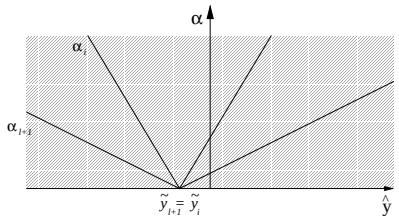

Figure d. $\mathbb{R}$

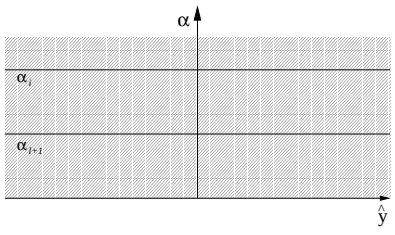

Figure e. $\mathbb{R}$

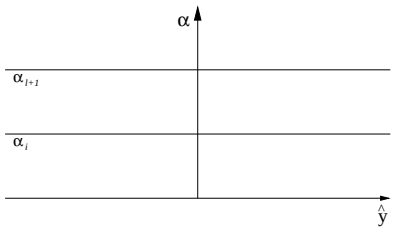

Figure f. $\emptyset$

Figure 1. Different possibilities for S_i

As we are interested in $|a_i + b_i\hat{y}|$ we can assume $b_i \geq 0$ for $i = 1, \ldots, l+1$ (if necessary, multiply both a_i and b_i by -1). If $b_i \neq b_{l+1}$ then α_i and α_{l+1} are equal at two points:

$$\frac{a_i - a_{l+1}}{b_{l+1} - b_i} \quad \text{and} \quad -\frac{a_i + a_{l+1}}{b_i + b_{l+1}} \tag{7}$$

Let u_i be the minimum and v_i be the maximum of the two. If $b_i = b_{l+1} \neq 0$ then

$$u_i = v_i = -\frac{a_i + a_{l+1}}{2b_i} \tag{8}$$

unless $a_i = a_{l+1}$, in which case $S_i = \mathbb{R}$.

From Fig. 1 we can see that S_i can be given as:

$$S_i = \begin{cases} [u_i, v_i] & \text{if} \quad b_{l+1} > b_i \\ (-\infty, u_i\,] \cup [v_i, \infty) & \text{if} \quad b_{l+1} < b_i \\ [u_i, \infty) & \text{if} \quad b_{l+1} = b_i > 0 \\ & \text{and} \quad a_{l+1} < a_i \\ (-\infty, u_i] & \text{if} \quad b_{l+1} = b_i > 0 \\ & \text{and} \quad a_{l+1} > a_i \\ \mathbb{R} & \text{if} \quad b_{l+1} = b_i = 0 \\ & \text{and} \quad |a_{l+1}| \leq |a_i| \\ \emptyset & \text{if} \quad b_{l+1} = b_i = 0 \\ & \text{and} \quad |a_{l+1}| > |a_i| \end{cases} \tag{9}$$

(the third and fourth lines follow from $\tilde{y}_i = -a_i/b_i$).

Now, we know that the p-value p can only change at the points u_i and v_i. So if $\hat{y}_1, \ldots, \hat{y}_{2l}$ are $u_1, \ldots, u_l, v_1, \ldots, v_l$ sorted in ascending order then p is constant on any interval $(\hat{y}_0, \hat{y}_1)$, $(\hat{y}_1, \hat{y}_2)$,..., $(\hat{y}_{2l-1}, \hat{y}_{2l})$, $(\hat{y}_{2l}, \hat{y}_{2l+1})$, where we defined $(\hat{y}_0 = -\infty$ and $\hat{y}_{2l+1} = \infty)$.

To calculate p for any of the above intervals, we count how many S_i include it and divide by the number of training examples plus one. If

$$N(\hat{y}_i) = \#\{S_j : [\hat{y}_i, \hat{y}_{i+1}] \subseteq S_j\} \quad \begin{matrix} i = 0, \ldots, 2l \\ j = 1, \ldots, l+1 \end{matrix} \tag{10}$$

then the p-value for the interval $[\hat{y}_i, \hat{y}_{l+1}]$ is $\frac{N(\hat{y}_i)}{l+1}$.

However, we said that we are more interested in finding labels with a particular p-value than finding p-values for particular labels. Our goal is to find a region of possible labels for which we can say that the probability of the true label lying outside that region is r or less; this is accomplished in Algorithm 1, which outputs the predictive region given the training set and a new example. The most computationally expensive step in the algorithm is the inversion of the ridge matrix $K + aI$, all other steps are at most $O(l^2)$.

Remark 1 Algorithm 1 usually gives almost precise tolerance regions, in the sense that (5) is near equality (indeed, all α_i will often be different, in which case (5) will be an equality for all $r = \frac{k}{l+1}, k \in \mathbb{N}$). These regions, however, may have 'holes' in them although our experiments show that these holes are rare in practice. Still, it might be preferable to modify Algorithm 1 re-

Algorithm 1 Ridge Regression Confidence Machine

Input training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, new example $\mathbf{x}_{l+1}$ and significance level r
Calculate $C = I - X(X'X + aI)^{-1}X'$, X being defined by (6)
Calculate $A = C(y_1, \ldots, y_l, 0)'$
Calculate $B = C(0, \ldots, 0, 1)'$
for $i = 1$ to $l + 1$ **do**
  calculate u_i and v_i as per (7) and (8)
end for
for $i = 1$ to $l + 1$ **do**
  find S_i as per (9)
end for
Sort $(-\infty, u_1, \ldots, u_{l+1}, v_1, \ldots, v_{l+1}, \infty)$ in ascending order obtaining $\hat{y}_0, \ldots, \hat{y}_{2l+3}$
Output $\cup_i [\hat{y}_i, \hat{y}_{i+1}]$, the union being over i such that $N(\hat{y}_i) > r$ (cf. (10))

placing its output by its convex closure.

5. Dual form ridge regression

If we use the well known dual formulation of ridge regression we can apply the kernel trick to find non-linear regression estimates. We rewrite our decision rule as

$$Y'(XX' + aI)^{-1}XX'$$

and so A becomes

$$(y_1, \ldots, y_l, 0)(I - (XX' + aI)^{-1}XX')$$

and B becomes

$$(0, \ldots, 0, 1)(I - (XX' + aI)^{-1}XX')$$

The kernel trick involves replacing dot products in input space with kernel functions which implicitly find dot products in some other (normally high-dimensional) space. We rewrite the dual form decision rule as

$$Y'(K + aI)^{-1}K$$

where K is a matrix of size $l + 1$ constructed using some kernel function $\mathcal{K}$:

$$K_{i,j} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \quad i = 1, \ldots, l + 1 \quad j = 1, \ldots, l + 1$$

A becomes

$$(y_1, \ldots, y_l, 0)(I - (K + aI)^{-1}K)$$

and B becomes

$$(0, \ldots, 0, 1)(I - (K + aI)^{-1}K)$$

We can then find p-values as with the primal form, the difference being that we have to invert a matrix of size $l + 1$ rather than d (the number of attributes).

The most computationally expensive part of our dual-form algorithm is therefore inverting the $l + 1 \times l + 1$ ridge matrix. We can, for example, use the Sherman-Morrison formula (Press et al., 1992) to speed this up.

6. Experimental results

In the experiments detailed below, 2 kernels were used, a radial basis function kernel

$$\mathcal{K}(u, v) = \exp -\frac{\sqrt{(u - v)'(u - v)}}{2\gamma^2}$$

with one parameter γ and a polynomial kernel

$$\mathcal{K}(u, v) = (u \cdot v + 1)^d$$

with one parameter d.

Figure 2 shows 1-dimensional examples generated with a modified version of the 'Mexican hat' function

$$y = 10\frac{\sin|x|}{|x|}$$

with x drawn randomly in the range $[-10, 10]$. Finally, Gaussian noise was added to the y values with 3 different variances, 1, 4 and 9.

90%, 95% and 99% confidence regions are shown based on a dual form ridge regression estimate made with a radial basis function kernel with $\gamma = 2$ and a set to 1. One can see how the confidence bounds adjust to the larger amounts of noise. The confidence bounds were tested by randomly sampling another 100 noisy points in the same manner as for the training set. Table 1 shows that the tolerance regions are 'nearly precise' in the sense of remark 1 for these testing points for all noise levels, even though the ridge regression estimate grows steadily worse as the noise is increased. The bounds are robust in the face of noisy data.

Figure 3 shows more examples based on the Mexican hat. Table 1 also details the results for these examples. Here, we vary the ridge parameter a. Again, the confidence regions remain 'nearly precise' even when changing the regression estimate's parameters result in much worse point prediction performance.

The method was also tested on several benchmark datasets from the UCI and DELVE repositories:

- auto-mpg. The data concerns fuel consumption in terms of miles per gallon. There are 392 examples, each with 7 attributes such as engine capacity and number of cylinders. 6 examples in

Table 1. Artificial dataset tolerance region performance

Noise	a	Target range	Mean |error|	% outside tolerance regions 90%	95%	99%	Mean width of tolerance regions 90%	95%	99%
$N(0,1)$	1	14.134	1.092	8.000	6.000	1.000	5.113	5.939	7.259
$N(0,4)$	1	20.551	1.891	9.000	6.000	1.000	7.384	8.249	13.220
$N(0,9)$	1	20.794	2.559	8.000	6.000	2.000	10.801	14.007	17.606
$N(0,1)$	0.1	14.059	0.913	12.000	6.000	3.000	3.478	4.210	5.875
$N(0,1)$	10	16.115	1.972	11.000	4.000	2.000	9.677	12.040	13.705
$N(0,1)$	100	14.376	2.601	11.000	6.000	1.000	14.591	17.343	19.520

the UCI repository were missing attribute values and were discarded. All examples originally had a unique identifying string which was discarded.

- Boston housing. The data concerns median house prices in \$1000s in suburbs of Boston. There are 506 examples, each with 13 attributes such as pollution and distances to various employment centers.
- Machine. This dataset concerns relative cpu performance. There are 209 examples each having 8 attributes such as cache memory size and cycle time.
- Servo. J. R. Quinlan says that 'It covers an extremely non-linear phenomenon — predicting the rise time of a servomechanism in terms of two (continuous) gain settings and two (discrete) choices of mechanical linkages.' There are 167 examples.

Following (Quinlan, 1993), each dataset was randomly split into approximately 10 equal parts. A kernel, kernel parameter and ridge factor for dual form ridge regression estimation were chosen for each dataset by cross-validating across the different parts and taking the values giving the smallest absolute error. The parameters are given in table 2. This experimental setup involves 'snooping' the data, we follow it only for the sake of comparison with Quinlan. For a more reliable comparison of ridge regression's predictive performance see (Saunders et al., 1998), here we are more interested in the validity of our tolerance regions.

Table 2. Parameters for dual ridge regression for benchmark datasets

Dataset	auto-mpg	housing	machine	servo
Kernel	rbf	rbf	poly	rbf
parameter	1.5	2.5	3	2.5
a	0.1	0.001	0.1	0.001

Dual form ridge regression has been shown to perform well on benchmark datasets before, here the method shows comparable performance to techniques previously used on the datasets. All results given here other than for ridge regression are taken from (Quinlan, 1993), more recently (Saunders et al., 1998) has shown comparable performance to support vector regression.

However, here we are interested in region prediction, not point prediction. The first interesting property of this algorithm is that, as stated in remark 1 it returns almost precise confidence regions. This means that as we randomly draw more and more training sequences and testing examples we would expect the percentage of testing examples lying outside predictive regions of significance r to approach $100(1-r)$. In table 4 we can see that the percentages are close to the expected values for our benchmark datasets.

The mean width of each tolerance region is given in table 4. Notice that the tolerance region widths for the machine dataset are much larger than the true label range of the dataset. This is due to a single outlier prediction by the underlying ridge regression estimate. We also give boxplots showing the range, median and quartiles of each confidence region's widths in figure 4. We can see that these more robust measures are much more reasonable for the machine dataset. Performance of region prediction algorithms can be compared with these widths; if the regions are 'nearly precise' as described above then the smaller the width the better.

7. Conclusion

This paper has described a shortcoming of many kernel based learning algorithms, the lack of confidence information associated with their predictions. It has presented a framework which can be applied to provide this information for these algorithms and detailed an efficient application of this framework for dual form ridge regression. Results on artificial datasets show that the confidence intervals generated by the technique are robust to changes in the amount of noise in the data and across different parameters for the un-

derlying regression. Results are also given for some benchmark datasets showing that the tolerance regions are 'nearly precise' for real world data. The paper suggests how to compare the tolerance regions' performance with other techniques. The approach allows 'nearly precise' tolerance regions to be found with no loss of prediction accuracy.

We would like to extend the framework to other popular regression algorithms such as support vector regression (the framework has been succesfully applied to SV pattern recognition (Saunders, 2000)). Ridge regression limits us to datasets of around 1000 examples, inverting larger matrices is impractical. More detailed comparisons with other frameworks such as Bayes and PAC would also be interesting.

References

Hoerl, A., & Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55–67.

Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (1992). *Numerical Recipes in C*. Cambridge University Press. 2 edition.

Quinlan, J. R. (1993). Combining Instance-Based and Model-Based Learning. *Proceedings ML '93*. San Mateo, CA: Morgan Kaufmann.

Saunders, C. (2000). *Efficient Implementation and Experimental Testing of Transductive Algorithms for Predicting with Confidence*. Doctoral dissertation, Royal Holloway and Bedford New College, University of London.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables. *Proceedings of the 15th International Conference on Machine Learning*.

Saunders, C., Gammerman, A., & Vovk, V. (1999). Transduction with Confidence and Credibility. *Proceedings of IJCAI '99*.

Vovk, V., & Gammerman, A. (2000). *Algorithmic theory of randomness and its applications in computer learning*. Unpublished manuscript.

Vovk, V., Gammerman, A., & Saunders, C. (1999). Machine-learning applications of algorithmic randomness. *Machine Learning, Proceedings of the Sixteenth International Conference (ICML'99)*.

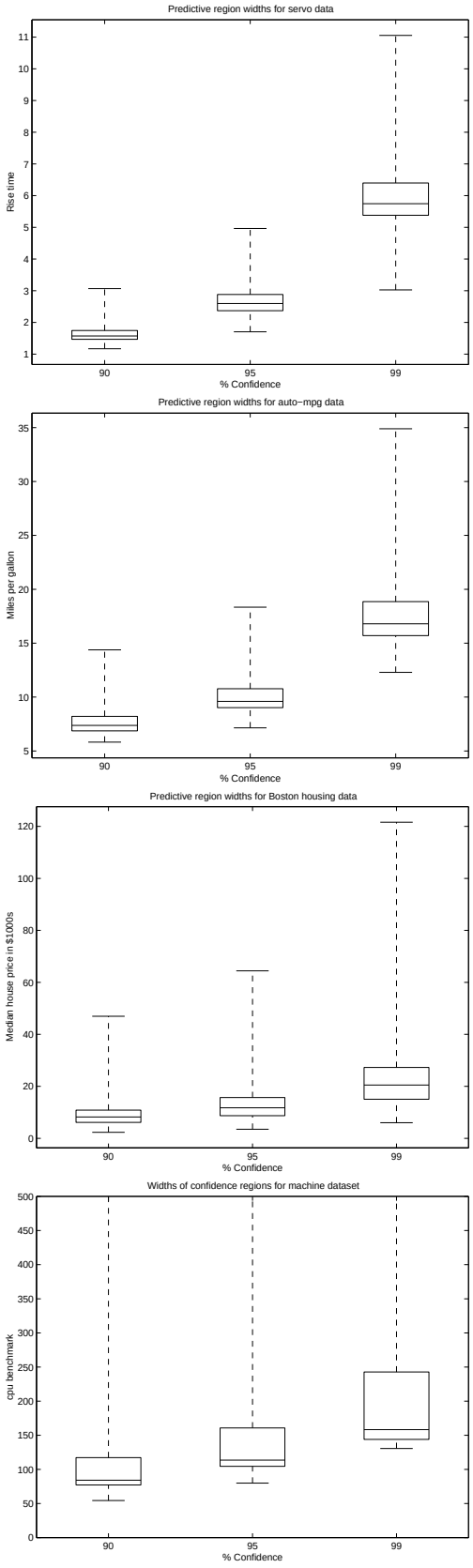

Figure 4. Distribution of tolerance region widths

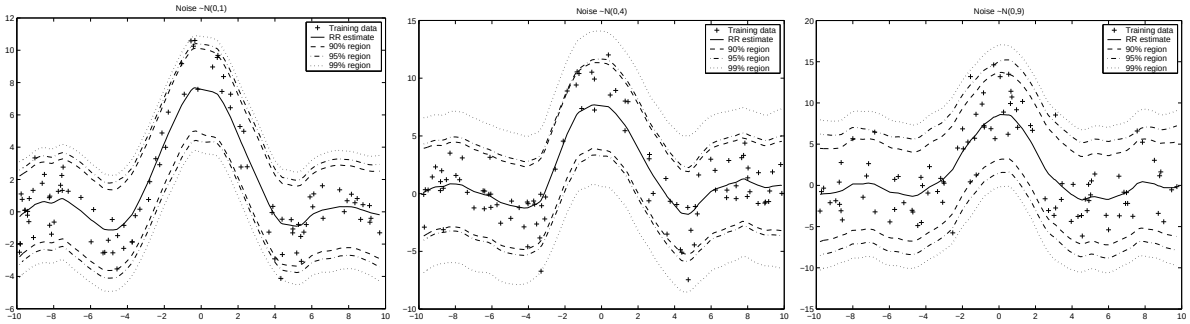

Figure 2. Ridge regression approximations to and typicalness confidence bounds for Mexican hat functions with different amounts of added noise

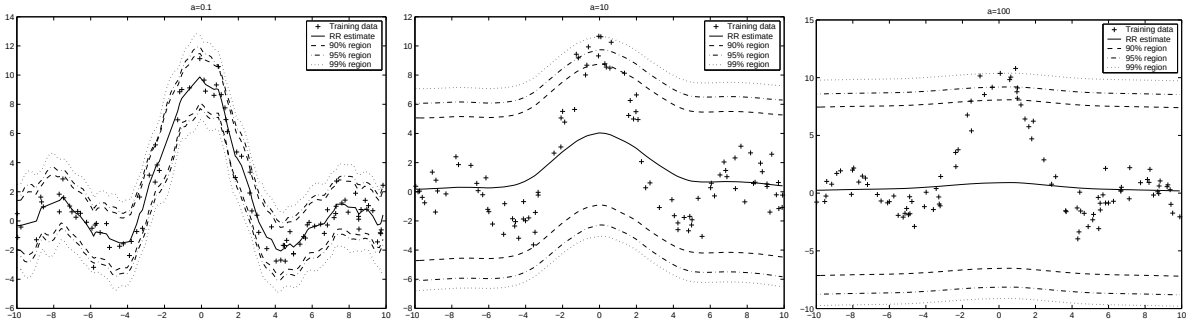

Figure 3. Ridge regression approximations to and typicalness confidence bounds for Mexican hat functions with different ridge factors

Table 3. Mean absolute error of regression predictions

Algorithm	Dataset			
	housing	machine	auto-mpg	servo
Instance-based learning	2.90	34.0	2.72	0.52
Linear regression	3.29	35.5	2.61	0.86
Linear regression + instances	2.45	30.0	2.37	0.48
Model trees	2.45	28.9	2.11	0.45
Model trees + instances	2.32	28.1	2.18	0.30
Neural nets	2.29	28.7	2.02	0.30
Neural nets + instances	2.23	29.0	2.06	0.29
Dual form ridge regression	1.85	28.6	1.83	0.32

Table 4. Tolerance region performance

	Target	Mean	% outside tolerance regions			Mean width of tolerance regions		
Dataset	range	|error|	90%	95%	99%	90%	95%	99%
machine	1144.000	28.630	10.048	6.220	1.435	51415.745	56209.581	60834.014
servo	6.969	0.318	8.982	5.389	1.198	1.638	2.655	5.954
housing	45.000	1.852	9.684	5.336	0.988	9.297	13.335	23.346
auto-mpg	37.600	1.827	10.459	4.592	1.020	7.708	10.066	17.627