The Equivalence between Row and Column Linear Regression:
A Surprising Feature of Linear Regression

Updated Version 2.0, October 2005

Volker Tresp
Siemens Corporate Technology
Department of Information and Communications
81730 München, Germany

Abstract

The rows of the design matrix in linear regression are the inputs of the training data set. Normal regression is row regression: the goal is to predict a training target from the corresponding row in the design matrix using a linear model. In contrast, in column regression we predict the components of the *test input* from the corresponding *columns* of the design matrix using a linear model and then use this model to predict the target of the test vector with the training targets as input. At first sight, column regression seems quite ridiculous. Surprisingly, both row and column regression give precisely identical predictions!

We discuss consequences of this feature for the sensitivity of regression towards regularization. In the context of collaborative filtering this result means that for linear regression, the prediction of customer preferences for certain items based on their preferences for other items is equivalent to the prediction of customer preferences for certain items based on the preferences of other customers for the same item.

1 Introduction

The training data set in linear regression consists of a set of input vectors and the corresponding targets. The parameters in the learned weight vector describe the linear influences of the inputs on the target. To obtain the prediction of the model for a new test input, one forms the inner product between the learned weight vector and the new input vector. We consider this to be *row linear regression*. Now let's consider the orthogonal problem and flip rows and columns. Now we form a linear model to predict

the components of the new *input* vector given the old input vectors. At first sight, this appears to be a rather strange idea since input vectors might be generated independently and might therefore be unpredictable from the input vectors in the training data set. We are now even more brave and, based on this linear model, try to predict the target of the new data point using as inputs the targets in the training data set. We consider this to be *column linear regression*. Surprisingly, both row and column linear regression lead to the same prediction.

In the Appendix we show that the equivalence also holds if no regularization is use but if the Moore-Penrose matrix inverse is used for calculating the weight vector.

An interesting consequence of the equivalence between row and column regression is that if either the row or the column problem is well defined, i.e. if there are either many more rows than columns or many more columns than rows, we obtain a prediction which is insensitive to the degree of regularization used (unless the model is very strongly regularized). As a consequence, the solution to a linear system with a large number of inputs and a small number of training data is well defined in the sense that the prediction is insensitive to the degree of regularization (unless the model is very strongly regularized).

This insight might be of importance in applications where both generative models underlying row and column regression make sense. Such an example is collaborative filtering where rows correspond to users and columns correspond to the items users have rated previously. With row regression, we would form a model to predict the probability that a user might be interested in a new item with the past ratings of this user as input. With column regression we would form a model to predict the probability that a user might be interested in a new product with the ratings of the other users for the new product as input. Both views seem to be reasonable. Based on the analysis in this paper, both approaches would lead to exactly identical predictions if one uses linear regression models. One would chose the approach best in terms of the practical constraints.

The main part of the paper is the following section where we introduce row and column linear regression and discuss their properties. We end with a concluding section.

2 Row and Column Linear Regression

2.1 Row Linear Regression

We start with a brief review of linear regression. Consider a training data set of N input vectors with dimension D, $\{x_i\}_{i=1}^N$ where $x_i = (x_{i,1}, \dots x_{i,D})^T$. x_i is the i-th row of the $N \times D$ design matrix X. y_i is the scalar target value associated with x_i. y is the N-dimensional vector of targets. We now want to form a linear model $x^T w$ to predict the target from the input. The least squares weight vector $w^{(LS)}$ minimizes the cost function

$$C^{(LS)} = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{D} x_{i,j} w_j \right)^2 = (y - Xw)^T (y - Xw).$$

Often one is interested to stabilize the solution and one adds a regularization term and uses the ridge regression cost function (Hastie, Tibshirani and Friedman, 2001)

$$C^{(RR)} = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{D} x_{i,j} w_j \right)^2 + \lambda \sum_{j=1}^{D} w_j^2 = (y - Xw)^T (y - Xw) + \lambda w^T w.$$

Here, $\lambda \geq 0$ modifies the degree of regularization. The minimizer is the regularized solution

$$w^{(RR)} = (X^T X + \lambda I)^{-1} X^T y$$

where I stands for the unity matrix.

Now, consider a new test input vector u. We predict for the new input u

$$\hat{t} = \sum_{j=1}^{D} u_j w_j^{(RR)} = u^T w^{(RR)}. \tag{1}$$

2.2 Column Linear Regression

Now we do something apparently unreasonable and try to predict the components of u from the *columns* of X, thus the new design matrix is X^T, the new target vector is u. Equivalently,

$$C^{(RR)} = \sum_{j=1}^{D} \left(u_j - \sum_{i=1}^{N} x_{i,j} v_i \right)^2 + \lambda \sum_{i=1}^{N} v_i^2 = (u - X^T v)^T (u - X^T v) + \lambda v^T v$$

is the regularized cost function.

We obtain as minimizer

$$v^{(RR)} = (XX^T + \lambda I)^{-1} X u. \tag{2}$$

Now we use this model to predict t with y as input and obtain as estimate

$$\tilde{t} = \sum_{i=1}^{N} y_i v_i^{RR} = y^T v^{RR}. \tag{3}$$

2.3 The Equivalence of Row and Column Regression

Consider the identity,

$$(XX^T + \lambda I)^{-1} X = X (X^T X + \lambda I)^{-1}. \tag{4}$$

Using this identity we obtain

$$\tilde{t} = y^T v^{RR} = y^T (XX^T + \lambda I)^{-1} X u = y^T X (X^T X + \lambda I)^{-1} u = u^T (X^T X + \lambda I)^{-1} X^T y$$
$$= t^T w^{RR} = \hat{t}$$

where we have used that $\hat{t} = \hat{t}^T$ since t is a scalar.

This result shows that indeed the predictions of row and column regression are identical.

2.4 Primal and Dual Regression

The solution of column regression, Equation 3, can also be written as (using $\tilde{t} = \tilde{t}^T$)

$$\tilde{t} = y^T v^{RR} = y^T (XX^T + \lambda I)^{-1} Xu = k^T (K + \lambda I)^{-1} y$$

where the Gram matrix K has components $(K)_{i,j} = x_i^T x_j$ and the vector k has components $(k)_i = u^T x_i$. This is the solution to the dual version of ridge regression (Saunders, C., Gammerman, Vovk, V., 1998). Thus, Equation 2 by itself is not surprising; *the surprising fact is that the solution to dual regression takes on exactly the form of column regression, as discussed.*

2.5 Discussion

What is really going on? Let's assume that row regression corresponds to the true data-generating process. In column regression we now form a model to predict the components of the input vector u from the columns of X and v_i^{RR} indicates how well u can be predicted from x_i reflecting the similarity between x_i and u. The similarity between x_i and u also reflects the similarity between y_i and t. Somehow, against one's intuition, this mechanism also works when the input vectors x_i and u where generated independently in which case the asymptotic ("true") $\{v_i^{RR}\}_{i=1}^D$ with $D \to \infty$ would be zero. The fact that the $\{v_i^{RR}\}_{i=1}^D$ are not zero is then, in this sense, only an effect of finite "sample", i.e. a finite D.

2.6 A Consequence for Collaborative Filtering

As discussed earlier, for collaborative filtering both row and column generative models are sensitive. Consider that $x_{i,j}$ is the rating of user i for item j and u_j is the rating of the user of interest for items i. y_i is the known rating of user i for a given new item. For the user of interest with ratings u we would like to predict the user's interest in the new item. One might now form a linear model to predict the ratings for the new item based on the ratings on items 1 to D (the row model) or a linear model to predict the rating of the new user based on the ratings of other users (the column model). Our theory tells us that both models make identical predictions..

2.7 Consequence for Regularization

Consider that $N >> D$ in which case row regression is well defined and insensitive to regularization. According to our results, column regression which has many fewer equations than variables is also well defined as long as a minimal degree of regularization is used and as long as the degree of regularization is not unreasonable large. Consequently, row and dual regression are both well defined when either $D << N$ or $N << D$. Only

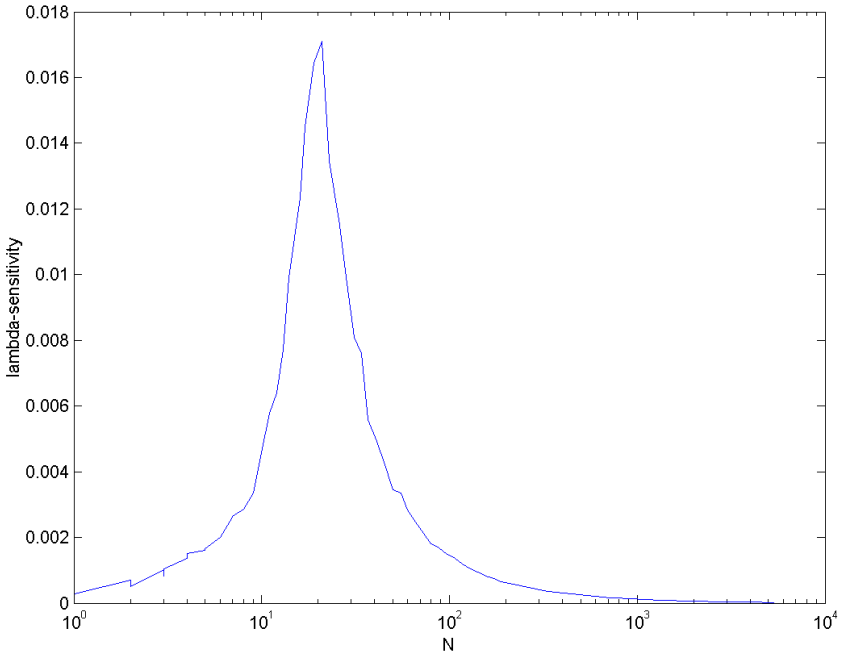

Figure 1: λ-sensitivity of the prediction error as a function of the training set size N. In the experiment, N random input vectors with dimension $D = 20$ were generated. The target y is a linear function of the inputs plus random noise. For each data set size, regularized linear models are trained where the regularization parameter λ is varied between 0.1 and 1.2. The λ-sensitivity is defined as $\frac{\max_\lambda(\text{test-error}(\lambda)) - \min_\lambda(\text{test-error}(\lambda))}{\max_\lambda(\text{test-error}(\lambda))}$. The plot shows the λ-sensitivity as a function of N. One sees that the test set error is most sensitive if $N \approx D$ and is rather insensitive if $N << D$ or $N >> D$.

when $D \approx N$, a fine-tuning of the regularization parameter λ is required, a fact well-known from practice.[1] Figure 1 illustrates this point.

2.8 Column Regression is not Completely Identical to Row Regression

Primal and Dual regression are mathematically identical. In particular they give the same estimate of the variance in the prediction.

$$var(\hat{t}) = \sigma_y^2 u^T (X^T X + \lambda I)^{-1} X^T X (X^T X + \lambda I)^{-1} u$$

$$= \sigma_y^2 u^T X^T (X X^T + \lambda I)^{-1} (X X^T + \lambda I)^{-1} X u.$$

[1] More precisely, we should compare D to the effective number of free parameters

The predicted variance of the column regression is different,

$$var(\hat{t}) = \sigma_u^2 y^T (XX^T + \lambda I)^{-1} XX^T (XX^T + \lambda I)^{-1} y.$$

Here, we assume i.i.d. target noise with variance σ_y^2, respectively σ_u^2. Since σ_y^2 and σ_u^2 are unrelated, the predicted variance cannot be identical in both cases.

This illustrates that the equivalence between row and column regression is more or less a remarkable coincidence.

3 Conclusions

The equivalence between row and column regression is quite surprising. We indicated potential consequences to collaborative filtering and to regularization. The equivalence does not hold for nonlinear systems. Here, the sensible combination of row and column regression might be an interesting area of future work.

4 Appendix

(With he help of an anonymous reviewer)

The equivalence is also valid if no regularization is used and one exploits the Moore-Penrose. Let $X^\dagger$ be the Moore-Penrose matrix inverse of X. Then we obtain as weight vector for row regression

$$w^{(Penrose)} = X^\dagger y$$

and the prediction is

$$\tilde{t} = u^T w^{(Penrose)} = u^T X^\dagger y$$

On the other hand we obtain for column regression

$$v^{(Penrose)} = \left(X^T\right)^\dagger u$$

and the prediction is

$$\tilde{t} = y^T v^{(Penrose)} = y^T \left(X^T\right)^\dagger u$$

Since

$$\left(X^T\right)^\dagger = \left(X^\dagger\right)^T$$

the identity follows.

5 References

Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of statistical learning. Springer, New York.

Saunders, C., Gammerman, Vovk, V. (1998). Ridge regression learning algorithms in dual variables. Proc. 15th International Conf. on Machine Learning.