FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
NATURWISSENSCHAFTLICHE
FAKULTÄT

Linear Regression

Lecture “Mathematics of Learning” 2022

Andreas Bärmann
Friedrich-Alexander-Universität Erlangen-Nürnberg

WIRTSCHAFTS
MATHEMATIK

Linear Regression

In a regression problem, targets are numeric values (quantitative).

- We will start from how we have performed a polynomial fit to data in the introduction to supervised learning: We have defined a polynomial function and found that it is only *linear* in the unknown coefficients.
  The coefficients that minimize a quadratic loss function could easily be derived via critical points.
  We now generalize the approach to higher-dimensional problems and to more general basis functions in a straight-forward way.

Linear Models and Least Squares

(see book Hastie et al. Elements of Statistical Learning, chapters 2,3.)
For some real input vector $X^\top = (X_1, X_2, \ldots, X_M)$, we want to predict a real-valued target output Y. We assume first that Y is a single value, i.e. Y is one-dimensional.
In order to predict the targets, we learn the coefficients of a function $f(X)$ with the help of a set of training data.
Then, for unseen data X, the target output Y is predicted by inserting X into the learned function: $Y = f(X)$.
The details: We model $f(X)$ as

$$(Y =) f(X) = \beta_0 + \sum_{j=1}^{M} X_j \beta_j$$

β_0: intercept, so-called *bias* in machine learning
This formula can also model polynomials via $X_2 = X_1^2, X_3 = X_1^3$, etc. (and even more general non-linear functions.)

Linear Models and Least Squares

In order to write the model compactly, we include a constant variable of value 1 in the matrix X, include β_0 in the vector of unknown coefficients β, and write the linear model for the unknown coefficients in vector form as

$$Y = X^\top \beta.$$

In M-dimensional input-output space, (X, Y) is a hyperplane. As β_0 is included in X, the hyperplane includes the origin and is a subspace.
Let set of training data $(x_1, y_1) \dots (x_N, y_N)$ with $x_i \in \mathbb{R}^M, y_i \in \mathbb{R}^K$ be given from which we estimate β.
Typically, the quality of the fit to a set of training data is measured by a quadratic loss function. This is also called *method of least squares*.
We thus determine the unknown coefficients β such that they minimize the *residual sum of squares RSS*

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} (y_i - x_i^\top \beta)^2$$

Linear Models and Least Squares

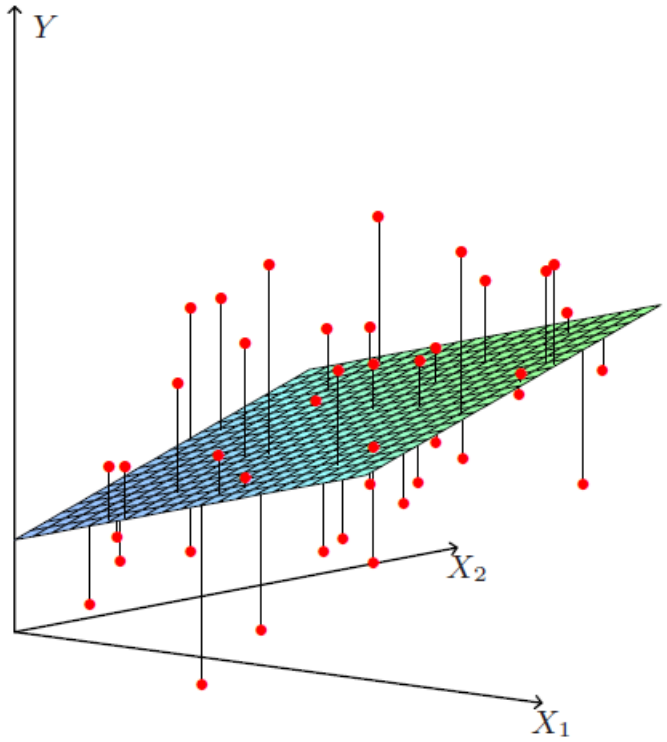

FIGURE 3.1. *Linear least squares fitting with* $X \in \mathbb{R}^2$. *We seek the linear function of* X *that minimizes the sum of squared residuals from* Y.

Linear Models and Least Squares

$RSS(\beta)$ is a quadratic function in $M+1$ parameters that can be written in matrix form as

$$RSS(\beta) = (y - \mathbf{X}\beta)^\top (y - \mathbf{X}\beta).$$

Taking partial derivative w.r.t. β yields: $\frac{\partial RSS}{\partial \beta} = -2\mathbf{X}^\top (y - \mathbf{X}\beta)$

We calculate the Hessian: $\frac{\partial^2 RSS}{\partial \beta \partial \beta^\top} = 2\mathbf{X}^\top \mathbf{X}$. We assume for the moment that $\mathbf{X}$ has full column rank. Hence, $\mathbf{X}^\top X$ is positive definite.

We determine critical points: $-\mathbf{X}^\top (y - \mathbf{X}\beta) = 0$

- From the start, we have inserted the data in a matrix $\mathbf{X}$. From this, we obtain the unique solution $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top y$.
- Fitted surface $f(X) = X^\top \hat{\beta}$ is fully characterized by $\hat{\beta}$.
- If new data point X comes in, target / output Y is predicted via $Y = X^\top \hat{\beta}$.

Linear Models and Least Squares

Suppose the data is such that columns of **X** are not linearly independent and so **X** does not have full rank, e.g. because of correlations. Non-full rank occurs often due to redundant coding.
Usually: We can do some preprocessing such that redundant columns in **X** are deleted. That is, we can assume that full rank is given.
The method can very easily be generalized to multidimensional target (output) vectors Y. (left as exercise).

Shrinkage Methods in Linear Models: Ridge Regression

High variabilities in regression results may occur. More stable methods additionally use some size reduction in the regression coefficients (see also regularization example in polynomial fit)

Shrinkage Methods in Linear Models: Ridge Regression

High variabilities in regression results may occur. More stable methods additionally use some size reduction in the regression coefficients (see also regularization example in polynomial fit)
Let us define least-square minimization problem for determining the best coefficients:

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{M} \beta_j^2\}. \quad (1)$$

λ: complexity parameter, controls the amount of shrinking.

Shrinkage Methods in Linear Models: Ridge Regression

High variabilities in regression results may occur. More stable methods additionally use some size reduction in the regression coefficients (see also regularization example in polynomial fit)
Let us define least-square minimization problem for determining the best coefficients:

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{M} \beta_j^2\}. \quad (1)$$

λ: complexity parameter, controls the amount of shrinking.
Determine minimizer $\hat{\beta}^{\text{ridge}}$:
Write argument from (1) in matrix form (w.l.o.g. $\beta_0 = 0$):

$$RSS(\lambda) = (\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^\top\beta$$

Alternative Derivation of Ridge Regression

$$RSS(\lambda) = (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) + \lambda \beta^\top \beta$$

best possible solution is given for

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

with $M \times M$ identity matrix $\mathbf{I}$. (please check...!)
If quadratic penalty $\beta^\top \beta$ is used, ridge regression solution is again linear in $\mathbf{y}$.
(compare with last week's formula for linear regression: $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top y$.)

Alternative Derivation of Ridge Regression

$$RSS(\lambda) = (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) + \lambda \beta^\top \beta$$

best possible solution is given for

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

with $M \times M$ identity matrix $\mathbf{I}$. (please check...!)
If quadratic penalty $\beta^\top \beta$ is used, ridge regression solution is again linear in $\mathbf{y}$.
(compare with last week's formula for linear regression: $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top y$.)
We see:

- Before inversion, positive constant $\beta^\top \beta$ is added on diagonal of $\mathbf{X}^\top \mathbf{X}$
- this makes resulting matrix *non-singular*, even if $\mathbf{X}^\top \mathbf{X}$ itself was singular. This is an advantage!

Brief Repetition Linear Algebra

Recall from linear algebra: *singular value decomposition (SVD)*
is factorization of a real or complex matrix, generalizes eigendecomposition.
SVD of a (not necessarily symmetric) matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ is a factorization of form $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^{\star}$, where

- $\mathbf{U}$ is $m \times m$ complex unitary matrix
- $\mathbf{D} \in \mathbb{R}^{m \times n}$ rectangular matrix with non-negative real numbers on the diagonal, otherwise zeros.
- $\mathbf{V}$ is $n \times n$ complex unitary matrix.
- for real matrix: $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^{T}$ with real orthonormal $\mathbf{U}, \mathbf{V}$.
- diagonal entries are called singular values
- SVD is not unique

Alternative Derivation of Ridge Regression

Write down the regression when $N \times p$ matrix $\mathbf{X}$ is decomposed as SVD: $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{N \times p}$, $\mathbf{V} \in \mathbb{R}^{p \times p}$ orthonormal matrics, columns of $\mathbf{U}$ span column space of $\mathbf{X}$, columns of $\mathbf{V}$ span row space.
$\mathbf{D} \in \mathbb{R}^{p \times p}$ diagonal matrix with entries $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$ *singular values* of $\mathbf{X}$.
Write down ridge solutions when $\mathbf{X}$ is decomposed by SVD:

$$\mathbf{X}\hat{\beta}^{\text{ridge}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} = \mathbf{U}\mathbf{D}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}\mathbf{U}^\top\mathbf{y} = \sum_{i=1}^{M} \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^\top \mathbf{y},$$

where $\mathbf{u}_j$ are columns of $\mathbf{U}$.
(This calculation is easy to verify, please double-check...)
We have $\lambda \geq 0$, thus $\frac{d_j^2}{d_j^2+\lambda} \leq 1$.
We see from this formula: Ridge regression computes the coordinates of (a new) target $\mathbf{y}$ for (new data) $\mathcal{X}$ with respect to the orthonormal basis $\mathbf{U}$.

Ridge Regression and PCA

Then it shrinks coordinates by factor $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Thus: more shrinking takes place if a coordinate has a basis vector with small d_j^2 when compared to a coordinate with large d_j^2.

Ridge Regression and PCA

Then it shrinks coordinates by factor $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Thus: more shrinking takes place if a coordinate has a basis vector with small d_j^2 when compared to a coordinate with large d_j^2.
Meaning of this: 'unimportant' columns are shrunk stronger.

Ridge Regression and PCA

Then it shrinks coordinates by factor $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Thus: more shrinking takes place if a coordinate has a basis vector with small d_j^2 when compared to a coordinate with large d_j^2.
Meaning of this: 'unimportant' columns are shrunk stronger.
Additional observation: SVD of $\mathbf{X}$ can express the *principal components* of $\mathbf{X}$, because:

Ridge Regression and PCA

Then it shrinks coordinates by factor $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Thus: more shrinking takes place if a coordinate has a basis vector with small d_j^2 when compared to a coordinate with large d_j^2.
Meaning of this: 'unimportant' columns are shrunk stronger.
Additional observation: SVD of $\mathbf{X}$ can express the *principal components* of $\mathbf{X}$, because: Let us denote the covariance matrix from test set by $\mathbf{S} := \frac{\mathbf{X}^\top \mathbf{X}}{N}$.

Ridge Regression and PCA

Then it shrinks coordinates by factor $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Thus: more shrinking takes place if a coordinate has a basis vector with small d_j^2 when compared to a coordinate with large d_j^2.
Meaning of this: 'unimportant' columns are shrunk stronger.
Additional observation: SVD of $\mathbf{X}$ can express the *principal components* of $\mathbf{X}$, because: Let us denote the covariance matrix from test set by $\mathbf{S} := \frac{\mathbf{X}^\top \mathbf{X}}{N}$.
We calculate $\mathbf{X}^\top \mathbf{X} = (\mathbf{V}^\top \mathbf{D}^\top \mathbf{U}^\top)\mathbf{U}\mathbf{D}\mathbf{V} = \mathbf{U}\mathbf{D}^2\mathbf{U}^\top$ is *eigendecomposition* of $\mathbf{X}^\top \mathbf{X}$ and of $\mathbf{S}$.

Ridge Regression and PCA

Then it shrinks coordinates by factor $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Thus: more shrinking takes place if a coordinate has a basis vector with small d_j^2 when compared to a coordinate with large d_j^2.
Meaning of this: 'unimportant' columns are shrunk stronger.
Additional observation: SVD of $\mathbf{X}$ can express the *principal components* of $\mathbf{X}$, because: Let us denote the covariance matrix from test set by $\mathbf{S} := \frac{\mathbf{X}^\top \mathbf{X}}{N}$.
We calculate $\mathbf{X}^\top \mathbf{X} = (\mathbf{V}^\top \mathbf{D}^\top \mathbf{U}^\top)\mathbf{U}\mathbf{D}\mathbf{V} = \mathbf{U}\mathbf{D}^2\mathbf{U}^\top$ is *eigendecomposition* of $\mathbf{X}^\top \mathbf{X}$ and of $\mathbf{S}$.
Recall: eigenvectors v_j are principal components of $\mathbf{X}$.
First eigenvalue direction v_1 leads to first principal component $\mathbf{z}_1 = \mathbf{X}v_1 = \mathbf{u}_1 d_1$.
Therefore: $\mathbf{u}_1$ is normalized first principal component, etc.
Property: first principal component has largest sample variance:
$Var(\mathbf{z}_1) = \frac{d_1^2}{N}$

Ridge Regression and PCA

Then it shrinks coordinates by factor $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Thus: more shrinking takes place if a coordinate has a basis vector with small d_j^2 when compared to a coordinate with large d_j^2.
Meaning of this: 'unimportant' columns are shrunk stronger.
Additional observation: SVD of $\mathbf{X}$ can express the *principal components* of $\mathbf{X}$, because: Let us denote the covariance matrix from test set by $\mathbf{S} := \frac{\mathbf{X}^\top \mathbf{X}}{N}$.
We calculate $\mathbf{X}^\top \mathbf{X} = (\mathbf{V}^\top \mathbf{D}^\top \mathbf{U}^\top)\mathbf{U}\mathbf{D}\mathbf{V} = \mathbf{U}\mathbf{D}^2\mathbf{U}^\top$ is *eigendecomposition* of $\mathbf{X}^\top \mathbf{X}$ and of $\mathbf{S}$.
Recall: eigenvectors v_j are principal components of $\mathbf{X}$.
First eigenvalue direction v_1 leads to first principal component $\mathbf{z}_1 = \mathbf{X}v_1 = \mathbf{u}_1 d_1$.
Therefore: $\mathbf{u}_1$ is normalized first principal component, etc.
Property: first principal component has largest sample variance: $Var(\mathbf{z}_1) = \frac{d_1^2}{N}$ in this order of components, variance gets smaller, last principal component has minimum variance.
Ridge regression shrinks these directions most. This ends the explanation about the relation of SVD and PCA.

Alternative Shrinking Model

Alternative (and potentially more restrictive) way of reducing coefficient sizes:

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2), \text{ s.t. } \sum_{j=1}^{M} \beta_j^2 \leq t\}.$$

parameter t: chosen beforehand, restricts size of coefficients
β_0 is left out from shrinking, as otherwise procedure would depend on origin

The Lasso Regression

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2), \text{ s.t. } \sum_{j=1}^{M}|\beta_j| \leq t\}.$$

w.l.o.g., $\beta_0 = 0$ (after centralizing data)
Write it in Lagrangian form as

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{M}|\beta_j|\}$$

.
We remark: quadratic ridge penalty expression of the form $\lambda\sum_{j=1}^{M}\beta_j^2$ is replaced by L_1 penalty $\lambda\sum_{j=1}^{M}|\beta_j|$ in the lasso regression which is nonlinear, however remains computationally tractable.

Remark

In practice, regression (and also other learning methods) are often solved via iterative so-called *gradient descent* methods. We will study them later.