Seminar in Data Science

Lecture 6: Regression

Laurent El Ghaoui

Seminar in Data Science and Information Technology, Summer 2020
TBSI – UC Berkeley

7/24/2020

Outline

Outline

What is regression?

In regression we are given a training set in the form of a matrix-vector pair:

$$X = [x_1, \dots, x_m] \in \mathbf{R}^{n \times m}, \quad y \in \mathbf{R}^m$$

where

- $x_i \in \mathbf{R}^n$ are m data points in n-dimensional “feature space”;
- $y = (y_1, \dots, y_m)$ are corresponding “outputs” or “responses”.

The goal of regression is to come up with a “prediction rule” $\hat{y}(x)$ that predicts the output for an unseen point $x \in \mathbf{R}^n$.

Linear and non-linear prediction

In linear prediction, we look for prediction rules of the form

$$\hat{y}(x) = w^T x + b$$

where $w \in \mathbf{R}^n$ and $b \in \mathbf{R}$ are the model parameters.

Most methods presented today are directly extended to "non-linear prediction rules", provided we work with non-linear features $\phi(x)$ instead of x, via

$$\hat{y}(x) = w^T \phi(x) + b.$$

Example:

$$\hat{y}(x) = w_1 x_1 + w_2 x_2 + w_3 x_1 x_2.$$

In a lecture 7 we explore these ideas in more detail; here we will focus on linear prediction rules.

Model fitting

To fit the model we usually solve a problem such as

$$\min_{w} \mathcal{L}(X^T w + b\mathbf{1}, y) + \lambda p(w),$$

where

- $\mathcal{L}$ is a convex loss function that encodes the error between the observed value and the predicted value;
- (w, b) are the model parameters;
- p is a penalty on the regression parameters;
- $\lambda > 0$ is a penalty parameter, obtained via cross-validation.

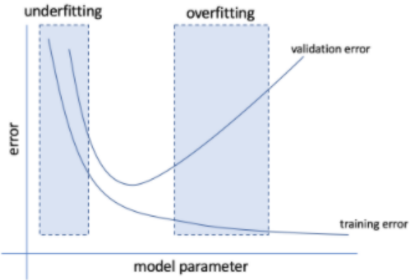
Most popular models are implemented in open-source packages such as scikit-learn [2].

Validation and testing

The cross-validation (over the penalty parameter λ) involves randomly selecting a subset of the data (representing say 70% of the data points), fitting the model, and testing on the remaining part via the prediction rule.

A new point is then given a predicted output via

$$\hat{y}(x) = w^T x + b.$$



Validation curve

Once that phase is done, we select the best value of the penalty parameter, and provide the final test results on an unseen test set.

Outline

Ordinary least-squares

Definition

Given $X \in \mathbf{R}^{n \times m}$, $y \in \mathbf{R}^m$, the *Ordinary Least-Squares* (OLS) problem is

$$\min_w \; \|X^T w - y\|_2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm, and $w \in \mathbf{R}^n$ is the variable.

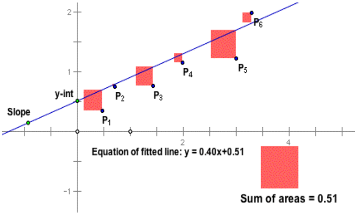
- Problem is ubiquitous ones in engineering, sciences, economics and finance.
- Solved by Legendre, Gauss ($\sim$ 1850).
- Very mature solution technology via linear algebra (*e.g.*, SVD) techniques.
- One of the most basic convex problems, used inside many convex optimization algorithms.

Least-squares
Applications

- Fitting auto-regressive models for log-return predictions.
- Various predictions in marketing, consumer credit, econometrics, etc.
- Solving simple portfolio optimization; index tracking.
- Generally, fitting models to data.

Interpretation

Smallest distance to consistency



OLS can be interpreted as finding the closest perturbation to "measurement" y to make equation $X^T w = y$ consistent (meaning, it has a solution w):

$$\min_{w,e} \|e\|_2 \;:\; X^T w = y + e.$$

e is noise that corrupted the measurement and made the model inconsistent.

Prediction

Previous interpretation useful in the context of prediction.

- In many cases, each column x_t of data matrix X corresponds to a measurement. (We use t to denote the column index.)
- The underlying model is

$$y_t = x_t^T w + e_t, \quad t = 1, \ldots, m,$$

where $e \in \mathbf{R}^T$ is a noise vector. Assume e is random, with $\mathbf{E}\, e = 0$.

Prediction

Previous interpretation useful in the context of prediction.

- In many cases, each column x_t of data matrix X corresponds to a measurement. (We use t to denote the column index.)
- The underlying model is

$$y_t = x_t^T w + e_t, \quad t = 1, \ldots, m,$$

  where $e \in \mathbf{R}^T$ is a noise vector. Assume e is random, with $\mathbf{E}\, e = 0$.

- *Question:* if we add one measurement (row x_{m+1}^T of X^T), what will be the corresponding output?

Prediction

Previous interpretation useful in the context of prediction.

- In many cases, each column x_t of data matrix X corresponds to a measurement. (We use t to denote the column index.)
- The underlying model is

$$y_t = x_t^T w + e_t, \quad t = 1, \ldots, m,$$

where $e \in \mathbf{R}^T$ is a noise vector. Assume e is random, with $\mathbf{E}\, e = 0$.

- *Question:* if we add one measurement (row x_{m+1}^T of X^T), what will be the corresponding output?
- *Answer:* since $\mathbf{E}\, e = 0$, the expected value of the new output y_{m+1} is

$$\hat{y}_{m+1} = x_{m+1}^T x.$$

Example

Prediction via auto-regressive models

Auto-regressive (AR) model for time-series y_t:

$$y_t = w_1 y_{t-1} + \ldots + w_n y_{t-n} + e_t, \quad t = 1, 2, 3, \ldots$$

where vector $w \in \mathbf{R}^n$ determines the model parameters.

Find x by fitting based on $n + p$ observations of past data $(y_t)_{t=1}^{t=n+m}$

$$\min_w \|X^T w - y\|_2,$$

where $y = (y_{n+m}, \ldots, y_{n+1})$, and $n \times m$ X has t-th column equal to $(y_{n+t}, \ldots, y_t)$, $1 \leq t \leq m$.

(Each column of X corresponds to a new time point.)

Example

Prediction via auto-regressive models

Once we've solved for w, we can make a prediction based on a new data value y_{n+m+1}:

$$\hat{y}_{n+m+1} = w_1 y_{n+m} + \ldots + w_n y_{m+1}.$$

Allows to form an average prediction error when we run the algorithm in a "sliding window" fashion.

Solution to OLS

If $p \times n$ matrix X is full row rank (XX^T is invertible), solution is unique:

$$w_{\mathrm{OLS}} = (XX^T)^{-1} Xy.$$

- Closed-form expression is rarely used. Algorithms such as QR decomposition or SVD are.
- Computational complexity grows as $\sim (nm^2 + m^3)$.
- Expression fails when X is not full rank. Then, nullspace of X^T describes ambiguity in solution. SVD methods can provide the whole subspace of solutions.

Regularized least-squares
Definition

In practice, OLS may provide solutions that are very sensitive to changes in input data (A, y).

Regularized LS:

$$\min_w \|X^T w - y\|_2^2 + \lambda \|w\|_2^2$$

where $\lambda > 0$ is the *regularization* parameter.

Stochastic interpretation:

$$\min_w \mathbf{E}\,\|(X + N)^T w - y\|_2^2$$

where N is random noise matrix, with $\mathbf{E}\,N = 0$ and $\mathbf{E}\,N^T N = \lambda I$.

Regularized least-squares

Solution

Solution: always unique, and given by

$$w_{\mathrm{RLS}} = (\lambda I_n + XX^T)^{-1} Xy.$$

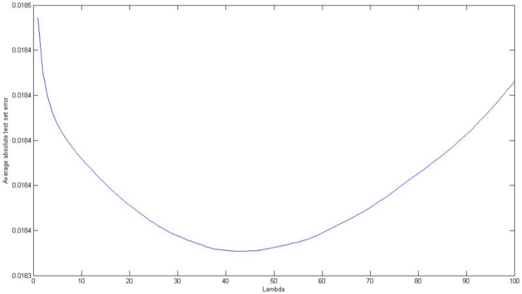
- Parameter $\lambda > 0$ enforces invertibility.
- This parameter is usually chosen via cross-validation.
- Again, closed-form expression rarely used; linear algebra techniques use OLS method for the equivalent (OLS) problem

$$\min_w \left\| \begin{pmatrix} X^T \\ \sqrt{\lambda} I_n \end{pmatrix} w - \begin{pmatrix} y \\ 0 \end{pmatrix} \right\|_2 .$$

(Note that matrix involved is always full rank, not matter what data X is.)

Example

AR model for prediction



AR model for prediction via regularized LS: average prediction error vs. regularization parameter.

- *Data:* APPL log-returns.
- *Method:* AR model fitted via regularized LS.
- Curve shows average prediction error, with algorithm run in "sliding window" mode.

Outline

Motivation

We will examine different models based on a linear assumption: that, for a new data point $x \in \mathbf{R}^n$, the predicted value is an *affine* function of the input x:

$$\hat{y}(x) = x^T w + b.$$

where $w \in \mathbf{R}^n$ contains the *regression coefficients* and $b \in \mathbf{R}$ is an offset. (In lecture 7, we explore non-linear alternatives.)

Together, w, b are the parameters of the model, which we wish to "learn" from training data samples (x_i, y_i), $i = 1, \ldots, m$.

Generalized regression

We consider the problem

$$\min_{w} \mathcal{L}(X^T w + b\mathbf{1}, y) + \lambda p(w),$$

where

- $\mathcal{L}$ is a convex loss function that encodes the error between the observed value and the predicted value;
- (w, b) are the model parameters;
- p is a penalty on the regression parameters;
- $\lambda > 0$ is a penalty parameter.

When $\mathcal{L}(z, y) = \|z - y\|_2^2$, $p(w) = \|w\|_2^2$, we recover regularized least-squares.

Playing with loss functions and penalties

Changing loss functions allos to cover these types of regression methods:

- Least-absolute deviation: to be less senstive to outliers than LS;
- Quantile regression: to predict intervals of confidence;
- Chebyschev regression: to work with largest errors only;
- KL divergence: to fit probability models

Typical penalties allow to

- l_1-norm: to enforce sparsity;
- l_2-norm (often, squared): to control statistical noise and improve prediction error;
- sum-block norms enable to enforce whole blocks of w to be zero.
- l_∞ norm tend to encourage "grouping" (elements in w have same magnitude).

LASSO

In LASSO, we solve the problem

$$\min_{w} \|X^T w - y\|_2^2 + \lambda \|w\|_1 .$$

- Here the model encourages sparsity of the result, due to the term $\|w\|_1$ in the penalty.
- The motivation is to be able to *interpret* the results, by finding the features that are most “predictive”.
- In practice, we cross-validate the choices of λ. Alternatively: select features first by (pure) LASSO, then run regularized LS. Another alternative is seen next.

LASSO can be unstable (non-unicity of the result), esp. with correlated features.

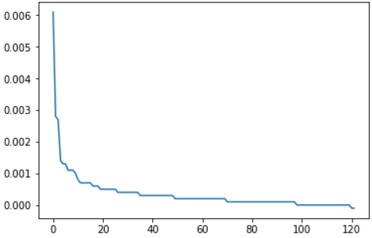
Elastic net

In Elastic net, we solve the problem

$$\min_{w} \|X^T w - y\|_2^2 + \lambda \|w\|_1 + \mu \|w\|_2^2,$$

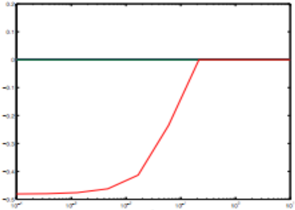
with $\mu > 0$ an extra regularization parameter.

- Here the model still encourages sparsity of the result, due to the term $\|w\|_1$ in the penalty.
- But it balances the sparsity against some stability.
- And allows for a better control of sparsity.
- Has the effect of grouping features together, which may be useful in its own right.

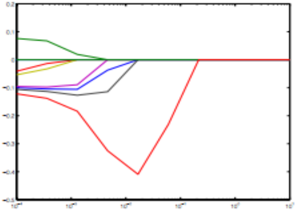


Largest features in w, approximated by two digits accuracy. Horizontal segments correspond to grouped features.

Controlling for sparsity



(a) Non-robust rank-1 square-root LASSO.



(b) Robust rank-1 square-root LASSO.

When data is low-rank, controlling sparsity is hard.

Robust regression: least-absolute deviation

In robust statistics, the goal is to handle outliers.

In least-absolute deviation, we solve the problem

$$\min_w \|X^T w - y\|_1 + \lambda p(w)$$

with (for example) $p(w) = \|w\|_2^2$.

- Since the l_1 allows some elements of the vector $X^T w - y$ to be large, it can tolerate **outliers** better than l_2-norm loss.
- This method is robust, but unstable: it may change much in result to changes in the data.
- Adding a (squared) regularization term $p(w) = w^T w$ allows to control unstability.
- An alternative is to use the Huber loss, seen next.

Robust regression: Huber loss

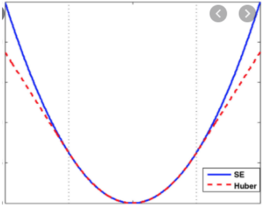
The Huber model is

$$\min_{w} \sum_{i=1}^{m} H_\delta(x_i^T w - y_i) + \mu \|w\|_2^2,$$

where H_δ is the so-called Huber function: for $z \in \mathbf{R}$,

$$H_\delta(z) = \begin{cases} \frac{1}{2} z^2 & \text{if } |z| \leq \delta \\ \delta |z| - \frac{1}{2}\delta^2 & \text{otherwise,} \end{cases}$$

with $\delta > 0$ a hyper-parameter.

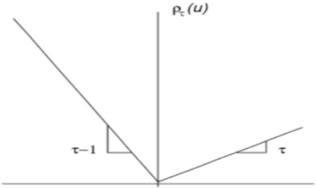


- Useful to handle outliers in data: large errors are handled by l_1-norm type of loss; small ones by a squared loss.
- Blends l_1 and l_2 penalties in a non-additive way.

Quantile regression: sample quantile

Given values $z_1, \dots, z_m$, the *median* is given by

$$\text{median}(z) = \arg\min_q \sum_{i=1}^{m} |z_i - q|.$$



More generally, the minimizer for the problem

$$\min_q \,(1-\tau) \sum_{z_i<q} (q - z_i) + \tau \sum_{z_i \geq q} (z_i - q) = \sum_{i=1}^{n} \rho_\tau(z_i - q),$$

gives the τ% quantile, with

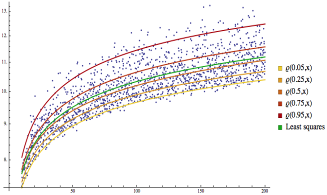
$$\rho_\tau(u) := \max(\tau u, (\tau - 1)u).$$

Quantile regression

In quantile regression, we solve the problem

$$\min_{w} \sum_{i=1}^{m} \rho_\tau(x_i^T w - y_i) + \lambda p(w)$$

with (for example) $p(w) = \|w\|_2^2$.



Quantile and least-squares regression on synthetic data.

- A linear or quadratic programming problem.
- Included in the most machine learning packages.

Outline

References

T. Hastie, R. Tibshirani, and J.H. Friedman.
The elements of statistical learning.
Springer, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay.
Scikit-learn: Machine learning in Python.
Journal of Machine Learning Research, 12:2825–2830, 2011.