Linear Regression

Dit-Yan Yeung

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

COMP 4211: Machine Learning (Fall 2022)

Regression Revisited

- Given a training set $\mathcal{S} = \{(\mathbf{x}^{(\ell)}, \mathbf{y}^{(\ell)})\}_{\ell=1}^{N}$ of N labeled examples each of which is in the form of an input-output pair.
- The problem is to estimate the parameters $\mathbf{w}$ in a regression function $f(\mathbf{x}; \mathbf{w})$ using $\mathcal{S}$ such that the predicted output $f(\mathbf{x}^{(\ell)}; \mathbf{w})$ for each input $\mathbf{x}^{(\ell)}$ is (usually) close to the actual output $\mathbf{y}^{(\ell)}$. Moreover, we want this to hold also for unseen examples sampled from the same data distribution.
- When the output $\mathbf{y}$ (with the superscript ℓ dropped for notational simplicity) is multivariate (i.e., $\mathbf{y}$ is a vector), it is a multi-output regression problem.
- A more common form, which will be our focus here, is when $\mathbf{y}$ is univariate (i.e., $\mathbf{y}$ degenerates to a scalar). We denote the output by y instead.
- The input $\mathbf{x} = (x_1, \ldots, x_d)^\top$ is d-dimensional (usually $d > 1$).

Linear Regression Function

- In linear regression, the regression function is simply a linear function:

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_d x_d = \mathbf{w}^\top \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^\top \mathbf{w},$$

  where $\mathbf{w} = (w_0, \ldots, w_d)^\top$ denotes the parameter vector and $\tilde{\mathbf{x}} = (x_0 = 1, x_1, \ldots, x_d)^\top$ is the augmented input with the constant 1 introduced as an additional dimension x_0.
- The weight w_0 is called the bias term which serves as an offset.
- The learning problem is to find the best $\mathbf{w}$ according to some performance measure using the training set $\mathcal{S}$.

Squared Loss

- Like many (though not all) machine learning models, a common way to learn the parameters $\mathbf{w}$ of the linear regression function $f(\mathbf{x};\mathbf{w})$ is to define a loss function $L(\mathbf{w};\mathcal{S})$ as a function of $\mathbf{w}$ and then minimize $L(\mathbf{w};\mathcal{S})$ w.r.t. $\mathbf{w}$.
- The most common loss function for regression problems is the squared loss (a.k.a. quadratic loss):

$$
\begin{aligned}
L(\mathbf{w};\mathcal{S}) &= \sum_{\ell=1}^{N} \left(f(\mathbf{x}^{(\ell)};\mathbf{w}) - y^{(\ell)} \right)^2 \\
&= \sum_{\ell=1}^{N} \left(w_0 + w_1 x_1^{(\ell)} + \cdots + w_d x_d^{(\ell)} - y^{(\ell)} \right)^2 .
\end{aligned}
$$

A Special Case ($d = 1$) for Illustration

- Squared loss:

$$L(\mathbf{w}; \mathcal{S}) = \sum_{\ell=1}^{N} \left(w_0 + w_1 x_1^{(\ell)} - y^{(\ell)} \right)^2 .$$

- Since $L(\mathbf{w}; \mathcal{S})$ is quadratic in $\mathbf{w}$, the optimal solution $\hat{\mathbf{w}}$ that minimizes $L(\mathbf{w}; \mathcal{S})$ is unique and can be found in closed form using the method of least squares.
- Setting the derivatives of $L(\mathbf{w}; \mathcal{S})$ w.r.t. w_0 and w_1 to 0 gives two linear equations:

$$N w_0 + w_1 \sum_{\ell=1}^{N} x_1^{(\ell)} = \sum_{\ell=1}^{N} y^{(\ell)}$$

$$w_0 \sum_{\ell=1}^{N} x_1^{(\ell)} + w_1 \sum_{\ell=1}^{N} \left(x_1^{(\ell)} \right)^2 = \sum_{\ell=1}^{N} x_1^{(\ell)} y^{(\ell)}.$$

Linear System in Matrix Form

- The linear equations can be expressed as:

$$\mathbf{A}\mathbf{w} = \begin{bmatrix} N & \sum_\ell x_1^{(\ell)} \\ \sum_\ell x_1^{(\ell)} & \sum_\ell \left(x_1^{(\ell)}\right)^2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} \sum_\ell y^{(\ell)} \\ \sum_\ell x_1^{(\ell)} y^{(\ell)} \end{bmatrix} = \mathbf{b}.$$

- Least squares estimate expressed in closed form:

$$\hat{\mathbf{w}} = \mathbf{A}^{-1}\mathbf{b},$$

assuming that $\mathbf{A}$ is invertible.

General Case ($d \geq 1$)

- Let the input and output parts of the N examples be expressed as:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_d^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_d^{(2)} \\ \vdots & & & \\ 1 & x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}.$$

- The method of least squares gives a set of $d+1$ linear equations which can be expressed in matrix form as:

$$\mathbf{A}\mathbf{w} = (\mathbf{X}^\top\mathbf{X})\,\mathbf{w} = \mathbf{X}^\top\mathbf{y} = \mathbf{b}.$$

- Least squares estimate:

$$\hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y},$$

assuming that $\mathbf{X}^\top\mathbf{X}$ is invertible.

Alternative Derivation using Multivariable Calculus

- The squared loss can also be expressed as:

$$\begin{aligned} L(\mathbf{w}; \mathcal{S}) &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 \\ &= (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= \mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\, \mathbf{w} - 2\, \mathbf{y}^\top \mathbf{X}\, \mathbf{w} + \mathbf{y}^\top \mathbf{y}, \end{aligned}$$

  where $\|\mathbf{v}\| = (\sum_i v_i^2)^{1/2}$ denotes the L_2 norm of a vector $\mathbf{v}$.
- To minimize $L(\mathbf{w}; \mathcal{S})$, we apply multivariate calculus to differentiate $L(\mathbf{w}; \mathcal{S})$ w.r.t. $\mathbf{w}$ and set the derivative to the zero vector $\mathbf{0}$ to get:

$$\begin{aligned} 2\, \mathbf{X}^\top \mathbf{X}\, \mathbf{w} - 2\, \mathbf{X}^\top \mathbf{y} &= \mathbf{0} \\ (\mathbf{X}^\top \mathbf{X})\, \mathbf{w} &= \mathbf{X}^\top \mathbf{y}, \end{aligned}$$

  which is the same as the matrix form of the system of $d + 1$ linear equations obtained above.

Complexity Considerations

- The closed-form solution requires inverting $\mathbf{X}^\top\mathbf{X}$ which is a $(d+1)\times(d+1)$ matrix.
- When d is large, instead of resorting to a closed-form solution, an alternative approach is to estimate $\hat{\mathbf{w}}$ iteratively like the logistic regression model for classification (to be discussed in the next topic).

Nonlinear Extensions

- For solving more complicated problems, nonlinear regression functions are needed.
- Different approaches for nonlinear extension:
  1. Explicitly adding more input dimensions (which depend nonlinearly on the original input dimensions) and applying linear regression to the expanded input
  2. Applying a nonlinear regression function to the original input
  3. Implicitly transforming the original input nonlinearly to a new space and applying a linear model to the transformed input
- We consider the first approach here and leave the other two for some later topics.

Polynomial Regression

- One common approach is to introduce higher-order terms as additional input dimensions, e.g., $x_i^2, x_i x_j, x_i x_j^2 x_k$.
- For notational simplicity, we only consider the $d = 1$ case and write the (only) input dimension x_1 simply as x.
- Polynomial function of degree d:

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x + \cdots + w_d x^d = \mathbf{w}^\top \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^\top \mathbf{w},$$

where $\tilde{\mathbf{x}} = (1, x, \ldots, x^d)^\top$ is like the augmented input used for linear regression.

Other Additional Input Dimensions

- Besides adding higher-order terms, more general transformations of the original input dimensions may also be introduced as additional input dimensions.
- Very often, feature engineering that uses domain knowledge to define application-specific features is applied.
- The method of least squares for linear regression can also be used here (for both polynomial regression and more general extensions) to obtain a closed-form solution.
- For the subsequent discussions, we use the same generic formulation regardless of whether additional input dimensions are introduced.

Model Overfitting

- If the training set $\mathcal{S} = \{(\mathbf{x}^{(\ell)}, y^{(\ell)})\}_{\ell=1}^{N}$ is small compared to the number of parameters in the linear regression function $f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_d x_d$, overfitting may occur.
- When overfitting occurs, it is common to find large magnitudes (i.e., absolute values) in at least some of the parameters.
- One common solution to the overfitting problem is to prevent the parameters from growing excessively large in magnitude.

Regularization

- Regularization is an approach which modifies the original loss function by adding one or more penalty terms, called regularizers, that penalize large parameter magnitudes.
- Regularized loss function based on L_2 regularization (a.k.a. Tikhonov regularization):

$$\begin{aligned} L_\lambda(\mathbf{w}; \mathcal{S}) &= L(\mathbf{w}; \mathcal{S}) + \lambda \|\mathbf{w}\|^2 \\ &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \\ &= \mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\, \mathbf{w} - 2\, \mathbf{y}^\top \mathbf{X}\, \mathbf{w} + \mathbf{y}^\top \mathbf{y} + \lambda\, \mathbf{w}^\top \mathbf{w}, \end{aligned}$$

  where $\lambda > 0$ is called the regularization parameter.
- In practice, not regularizing w_0 usually gives better result because w_0 serves as an offset which is not multiplied with any input variable.

Closed-Form Solution with L_2 Regularization

- By differentiating $L_\lambda(\mathbf{w}; \mathcal{S})$ w.r.t. $\mathbf{w}$ and setting the derivative to the zero vector $\mathbf{0}$, we obtain:

$$\begin{aligned} 2\,\mathbf{X}^\top \mathbf{X}\,\mathbf{w} - 2\,\mathbf{X}^\top \mathbf{y} + 2\,\lambda\,\mathbf{w} &= \mathbf{0} \\ (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})\,\mathbf{w} &= \mathbf{X}^\top \mathbf{y}, \end{aligned}$$

  where $\mathbf{I}$ is the identity matrix.
- The least squares estimate can also be obtained in closed form:

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}.$$

- Note that $\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}$ is always invertible for any $\lambda > 0$.
- Linear regression with L_2 regularization (a.k.a. ridge regression) degenerates to the ordinary linear regression (without regularization) when $\lambda = 0$.

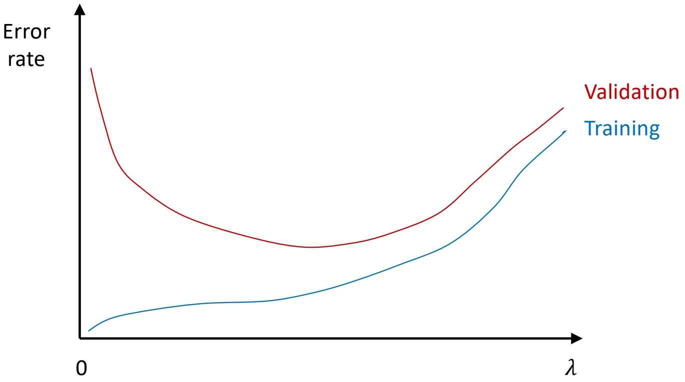
How to Set λ?

- Though not the only method, cross validation (or, more correctly, called holdout validation) is commonly used by training a model (using a regularized loss function with a specific value of λ) on a training set and validating the trained model on a separate validation set (which mimics the test set not available during model training).
- Different values of λ are used to obtain different least squares estimates. The value that gives the smallest error rate in the validation set is chosen.

Overfitting and Underfitting

- Suppose the training set is small compared to the number of model parameters.
- Overfitting:
  - May occur if λ is too small
  - Small training error
  - Large validation error
- Underfitting:
  - May occur if λ is too large
  - Large training error
  - Large validation error

Typical Training and Validation Error Curves



Other Regularizers

- Many other regularizers can also be defined.
- For example, instead of using the L_2 norm, the L_p norm for some other value of p (e.g., 1 or 0) has also been used:

$$\|\mathbf{v}\|_p = \Big(\sum_i |v_i|^p\Big)^{1/p}.$$

(The L_2 norm may also be written explicitly as $\|\cdot\|_2$.)

- Linear regression with L_1 regularization, also called LASSO (least absolute shrinkage and selection operator), favors sparse solutions with all but a small number of dimensions equal to 0.
- Although the L_1 norm is also convex like the L_2 norm, there is no closed-form solution for LASSO. Iterative algorithms are needed for estimating the parameters.

Mean Squared Error

- A common performance metric for regression problems is the mean squared error (MSE):

$$MSE = \frac{1}{N}\sum_{\ell=1}^{N}\left(f(\mathbf{x}^{(\ell)};\mathbf{w}) - y^{(\ell)}\right)^2,$$

  which is similar to the squared loss but with two differences:
  - MSE can be used for the validation set and test set in addition to the training set.
  - MSE measures the mean over all the examples in the set, not the sum.
- Instead of using MSE, it is more common to use the root mean squared error (RMSE), which is the square root of MSE, to bring it back to the same level of prediction error for easier interpretation.

R^2 Score

- Another commonly used performance metric for regression problems is the coefficient of determination or R^2 score:

$$R^2 = 1 - \frac{\sum_{\ell=1}^{N} \left(f(\mathbf{x}^{(\ell)}; \mathbf{w}) - y^{(\ell)}\right)^2}{\sum_{\ell=1}^{N} \left(\bar{y} - y^{(\ell)}\right)^2},$$

where

$$\bar{y} = \frac{1}{N} \sum_{\ell=1}^{N} y^{(\ell)}.$$

- The best possible R^2 score is 1 when the corresponding MSE is 0.
- When the model always predicts the mean value of y, the R^2 score will be equal to 0.
- Negative values are also possible because the model can have arbitrarily large MSE.