Predictive Modelling - III

Regression

Rita P. Ribeiro

Data Mining I - 2023/2024

U. PORTO
FC FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO
[dcc] DEPARTAMENTO DE CIÊNCIA DE COMPUTADORES FACULDADE DE CIÊNCIAS DA UNIVERSIDADE DO PORTO

Summary

- Regression
  - Problem Definition
  - Bias and Variance
  - Evaluation Metrics
- Linear Regression Methods

Regression

Regression: Problem Definition

Setting

- $D = \{\langle \mathbf{x_i}, y_i \rangle\}_{i=1}^{N}$
- $\mathbf{x_i}$ feature vector with p predictor variables
- $y_i \in \mathbb{R}$ target numerical variable Y
- There is an unknown function $Y = f(\mathbf{x})$

Goal: Learn the best approximation of the unknown function $f()$

Approach

- Approximate $f()$ by $h_\theta(\mathbf{x})$
- Follow a preference criterion over the parameterization space θ
- Search for the “best” $h()$ according to the criterion and the data set

Regression: Problem Definition

Regression Model

- A function that transforms a vector of values of the predictors, **x**, into a real number, y
- It assumes the following relationship

$$y_i = h_\theta(\mathrm{x}_i) + \varepsilon_i$$

where

- $h_\theta(\mathrm{x}_i)$ is a regression model with the set of parameters θ
- ε_i are observation errors (i.e. residuals)

Bias and Variance

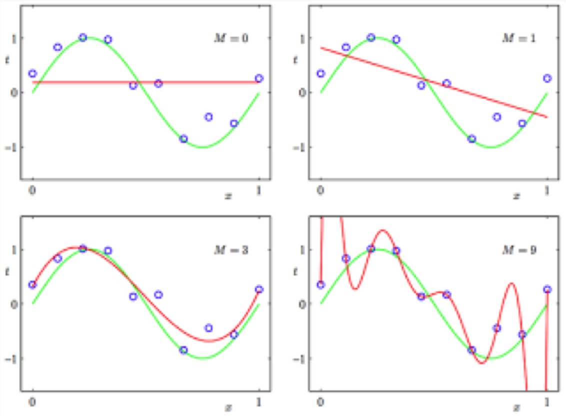
Bias-variance decomposition of the error helps:

- explain why simple learners can outperform powerful ones
- explain why model ensembles outperform single models
- understand and avoid overfitting

Bias and Variance

Avoid overfitting

- Polynomials of different orders M to fit the data.



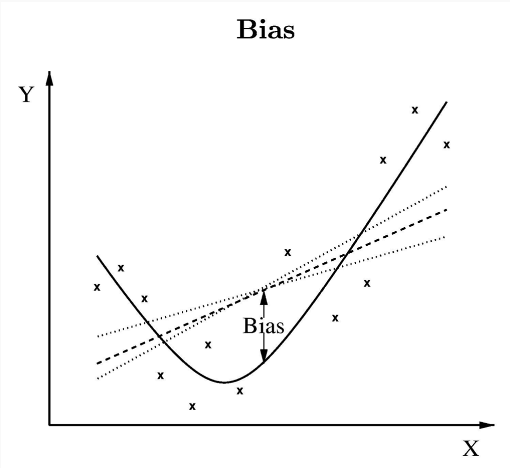
- Which one overfitts the data?

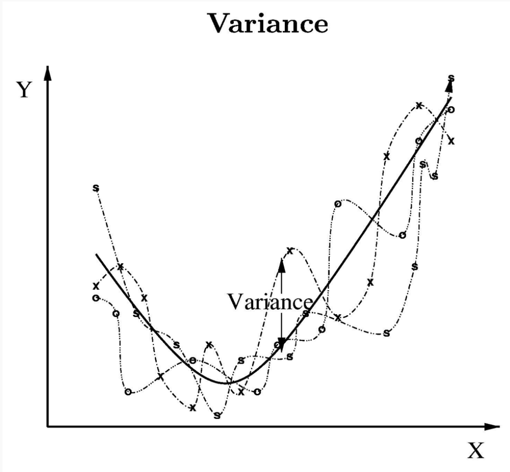
Bias and Variance

- Given a traning set $D = \{\langle \mathbf{x_i}, y_i \rangle\}_{i=1}^{N}$
- The learner induces a regression model $\hat{y} = h_\theta(\mathrm{x})$
- Loss functions measure the quality of learner's predictions
  - Squared loss: $L(y, \hat{y}) = (y - \hat{y})^2$
  - Absolute Loss: $L(y, \hat{y}) = |y - \hat{y}|$
  - Zero-one loss: $L(y, \hat{y}) = 0$ if $y = \hat{y}$, 1 otherwise
  - ...
- In the training set, we can obtain the Expected Loss, i.e. $E[L(y, \hat{y})]$

Bias and Variance

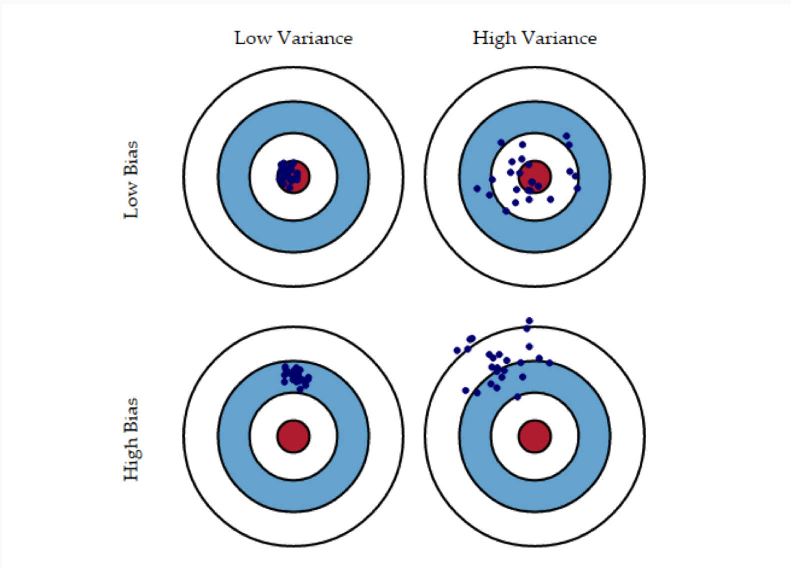
Expected Loss = Bias + Variance





Bias and Variance

Expected Loss = Bias + Variance



Bias and Variance

Bias-variance trade-off

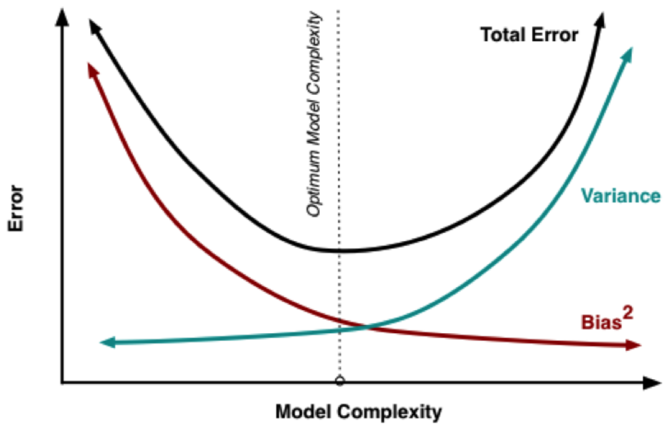


Fig. 6 Bias and variance contributing to total error.

Bias and Variance

What should $\hat{y}$ be?

- Prediction with minimum average loss relative to all predictions

$$\hat{y} = \underset{y'}{\text{argmin}}\, E[L(y, y')]$$

- for Squared Loss is the mean, i.e. $\hat{y} = \bar{y}$
- for Absolute Loss is the median, i.e. $\hat{y} = \tilde{y}$
- for Zero-one Loss is the mode

How to obtain reliable estimates of the error to compare models performance?

Evaluation Metrics

Evaluation Metrics

- Mean Squared Error (MSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

where

- $\hat{y}_i$ is the prediction of the model under evaluation for the case i
- y_i the respective true target variable value.

- It is measured in a unit that is squared of the original variable scale.
- Thus, it is common to use the Root Mean Squared Error $RMSE = \sqrt{MSE}$

Evaluation Metrics

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$$

where

- $\hat{y}_i$ is the prediction of the model under evaluation for the case i
- y_i the respective true target variable value.
- MAE is measured in the same unit as the original variable scale.

Evaluation Metrics

Relative Error Metrics

- Unit less metrics which means that their scores can be compared across different domains.
- They are calculated by comparing the scores of the model under evaluation against the scores of some baseline model.
- The relative score is expected to be a value between 0 and 1, with values nearer (or even above) 1 representing performances as bad as the baseline model, which is usually chosen as something too naive.

Evaluation Metrics

- A common baseline model is the constant model that predicts for all test cases the average target variable value ($\bar{y}$) calculated in the training data.
- Normalized Mean Squared Error (NMSE)

$$NMSE = \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\bar{y} - y_i)^2}$$

- Normalized Mean Absolute Error (NMAE)

$$NMAE = \frac{\sum_{i=1}^{N}|\hat{y}_i - y_i|}{\sum_{i=1}^{N}|\bar{y} - y_i|}$$

- Both vary between 0 an 1. The closer to 0, the better.

Evaluation Metrics

- Correlation Coefficient

$$\rho_{\hat{y},y} = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}}_i)^2 \sum_{i=1}^{N}(\hat{y}_i - \bar{y}_i)^2}}$$

- Varies between -1 and 1.
- Values between -0.8 and 0.8 are not, typically, considered relevant.

Evaluation Metrics

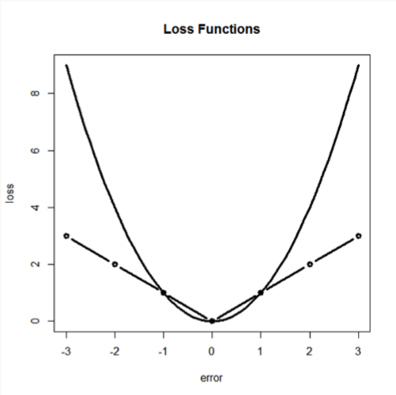
- Coefficient of determination - ratio R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

- Varies between 0 and 1.
- The closer to 1, the better.
- Gives the notion of the percentage of observed variation explained by the model.

Evaluation Metrics: Wrap-up

- *MSE*-related metrics amplify the large errors
  - It may be good in areas where large errors are intolerable.
- *MAE*-related metrics are not as sensitive to large errors
  - Treats all errors the same way
  - Gives a better indication of the “typical” error of the model



Evaluation Metrics: Wrap-up

- The relative measures (e.g. NMSE,NMAE) have the advantage of independence of the application domain.
- The correlation coefficient measures the strength of the relationship between the model output and the true target variable.
  - For multiple linear regression it is difficult to explain because we have multiple variables involved here.
- The coefficient of determination R^2 is indicative of the level of explained variability in the data set.
  - If $R^2 = 0.50$, then approximately half of the observed variation can be explained by the model
  - It is a convenient rescaling of MSE that is unit invariant

Linear Regression Methods

Predictive Modelling: Where we at?

- Distance-based Approaches
  - e.g. kNN
- Probabilistic Approaches
  - e.g. Naive Bayes, Bayesian Networks
- Mathematical Formulae
  - e.g. multiple linear regression
- Logical Approaches
  - e.g. CART
- Optimization Approaches
  - e.g. SVM, ANN
- Ensemble Approaches

Simple Linear Regression

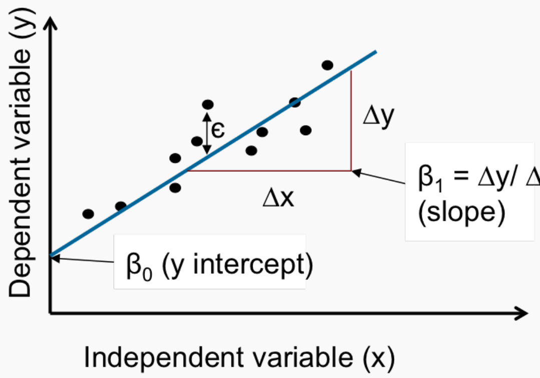
The very simplest case: one predictor variable x and one target variable y.

- The model is a straight line that approximates the relationship between the two, defined by

$$y_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i$$

where

- β_0 is the y intercept
- β_1 is the slope
- ε_i is the error for instance i



Multiple Linear Regression

- One of the approaches to the multiple regression problem
- The functional form of the regression model is

$$Y = \beta_0 + \beta_1 \cdot X_1 + \cdots + \beta_p \cdot X_p$$

- The goal is to find the vector of parameters β that minimizes the sum of the squared errors (SSE)

$$SSE = \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 \cdot X_1 + \cdots + \beta_p \cdot X_p))^2$$

- The minimization of SSE can be solved by $\beta = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$ or by using Singular Value Decomposition (SVD).

Multiple Linear Regression

Multicollinearity problem

- highly correlated predictor variables cause variance to be large, highly dependent on the training data
- model predictions become unstable

Regularization

- Tune the model to achieve a good bias-variance trade-off.
- Add a bias to the regression estimate to make sure that the coefficients are, on average, small in magnitude - *shrinkage*

Multiple Linear Regression

- *Ridge Regression*: shrinks the coefficients using least squares by adding the regularization term $\lambda \sum_i \beta_i^2$ (L_2 norm).

$$\sum_{i=1}^{N}(y_i - (\beta_0 + \beta_1 \cdot X_1 + \cdots + \beta_p \cdot X_p))^2 + \lambda \sum_i \beta_i^2$$

- *Lasso Regression*: shrinks the coefficients using least absolute values by adding the regularization term $\lambda \sum_i |\beta_i|$ (L_1 norm).

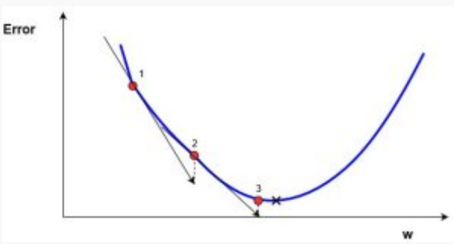
$$\sum_{i=1}^{N}(y_i - (\beta_0 + \beta_1 \cdot X_1 + \cdots + \beta_p \cdot X_p))^2 + \lambda \sum_i |\beta_i|$$

–>

Linear Regression using Gradient Descent

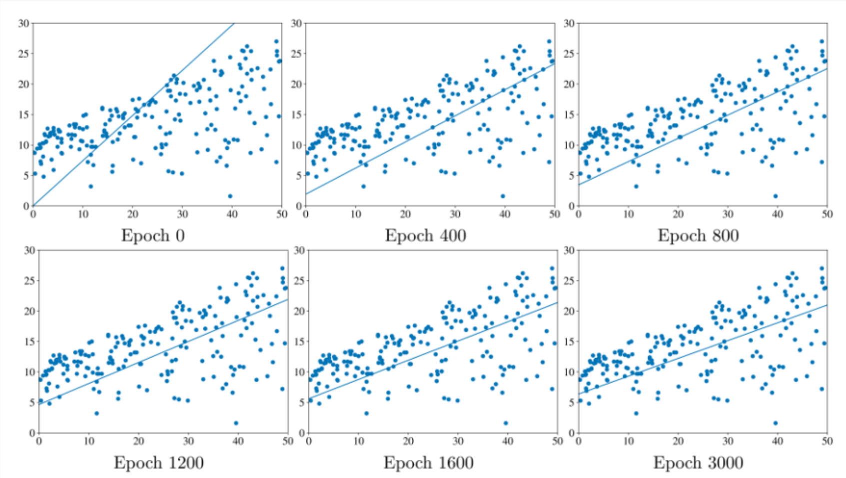
Gradient Descent

- An iterative optimization algorithm to find the minimum of a function.
- In case of regression the goal is to minimize the error function.
- In linear regression it calculates the partial derivative of the loss function w.r.t. to each coefficient and updates them until the loss reaches a very small value, ideally 0.



Linear Regression using Gradient Descent

- Gradient Descent operates by training epochs



- It can be slow to run on very large datasets.

Linear Regression using Gradient Descent

- Batch
  - calculates the error for each example in the training data and, only afterwards, updates the model
- Stochastic
  - calculates the error and updates the model for each example in the training data.
- Mini-batch
  - training data is split into small batches that are used to calculate the error and update the model.

Linear Regression: Wrap-up

Pros

- Well-known with many variants of this simple methodology
- Effective approach when the “linearity” assumption holds
- The model is intuitive - a set of additive effects of each variable towards the prediction
- Computationally very efficient

Cons

- Too strong assumptions on the shape of the unknown function

Note

- Techniques such as regularization, gradient descendent can be applied to other regression methods.

Other Regression Methods

- k-Nearest Neighbors
  - Predicts the average of the target variable values of the neighbors
- LOESS (Locally Estimated Scatterplot Smoothing)
  - Non-parametric method that combines multiple linear least squares regression models in a k-nearest neighbor-based way
- MARS (Multiple Additive Regression Splines)
  - Non-parametric method that extends linear regression, tackling nonlinearities and interactions between variables.

Other Regression Methods

- Support Vector Machines
- Artificial Neural Networks
- Random Forests
  - based on ensemble of CART trees
- eXtreme Gradient Boosting (XGBoost)
  - optimized distributed gradient boosting provided by parallel tree boosting
- Many more exist . . .

References

References

Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.

Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.

Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Moreira, João, Andre Carvalho, and Tomás Horvath. 2018. *Data Analytics: A General Introduction*. Wiley.

Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.

Torgo, Luís. 2017. "Data Mining i Course." Slides.