Machine Learning

Condensed Notes

Gabriel Sigurd Cabrera

stk-in4300 & fys-stk4155

August 26, 2019

Contents

CHAPTER 1

Introduction

Given a set of features $\{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_p\}$ we can construct a *learner* that will use an outcome $\mathbf{Y}$ in order to predict other potential outcomes. This is called supervised learning.

There are several types of outcomes, **quantitative**, **categorical**, and **ordered categorical** outcomes. The predicting task for a quantitative response is called **regression**, and for a categorical response it is called **classification**.

Quantitative outcomes are binary[1] in nature, while *categorical* outcomes can represent continuous or discrete non-binary values; *ordered categorical* outcomes are a subcategory of categorical outcomes in which the outcomes are interrelated on a scale[2].

It is common to have three datasets - a **training set**[3], a **test set**[4] and a **validation set**[5].

1.1 Two Prediction Methods

1.1.1 Generalized Linear Models and Least Squares

To predict a set of outputs $\hat{\mathbf{y}}$ based on a test input $\mathbf{X}$ we use the following model:

$$\hat{\mathbf{y}} = \hat{\mathbf{X}}^{\mathrm{T}}\hat{\beta}$$

Where $\mathbf{y}$ is an $(N \times O)$ matrix, $\mathbf{X}$ is an $(N \times p)$ matrix, and $\hat{\beta}$ is a $(p \times O)$ matrix, whose first row/element β_0 is called the **intercept** or **bias**. $\hat{\beta}$ itself is known as the **vector of coefficients**. This can also be visualized as follows:

$$\begin{bmatrix} \hat{y}_{1,1} & \hat{y}_{1,2} & \cdots & \hat{y}_{1,\text{O}} \\ \hat{y}_{2,1} & \hat{y}_{2,2} & \cdots & \hat{y}_{2,\text{O}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{\text{N},1} & \hat{y}_{\text{N},2} & \cdots & \hat{y}_{\text{N},\text{O}} \end{bmatrix} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,p} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{\text{N},1} & X_{\text{N},2} & \cdots & X_{\text{N},p} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \hat{\beta}_{1,1} & \hat{\beta}_{1,2} & \cdots & \hat{\beta}_{1,\text{O}} \\ \hat{\beta}_{1,1} & \hat{\beta}_{1,2} & \cdots & \hat{\beta}_{1,\text{O}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\beta}_{\text{N},1} & \hat{\beta}_{\text{N},2} & \cdots & \hat{\beta}_{\text{N},\text{O}} \end{bmatrix}$$

Where each *column* of the matrix $\mathbf{X}$ represents a **feature** of the dataset, and each *column* represents a single datapoint corresponding to an output in $\mathbf{y}$.

To find the vector of coefficients $\hat{\beta}$ we need a $(p \times N)$ set of inputs[6] $\mathbf{X}$ and their known outputs $\mathbf{y}$:

$$\hat{\beta} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{1.1}$$

[1] Meaning that they can be either *True*, or *False*

[2] This could be *small*, *medium*, and *large*.

[3] Used to train the algorithm to generate outputs in a specific way

[4] Hidden dataset, to which the learner is not exposed.

[5] Optional in some cases, used to validate previous results with a third hidden dataset.

[6] Keep in mind that $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ must be *nonsingular*!

Derivation of $\hat{\beta}$

To find an expression for the vector of coefficients, it is necessary to minimize the distance between each point in a given dataset, and a predicted line. This means we want to minimize something known as the **residual sum of squares**:

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^{\mathrm{T}}\beta)^2 \tag{1.2}$$

Recall that functions can be minimized via differentiation. We should rewrite equation 1.2 in vector form to accomplish this:

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\beta)$$

We then take the derivative with respect to the vector of coefficients and set this to zero, keeping in mind that we must redefine some variable in the equation to take this into account. We choose to put a hat on β:

$$0 = \mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\hat{\beta})$$

Solving for $\hat{\beta}$ is the final step:

$$\hat{\beta} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

Quick Notation Reference

Table 1.1: Dimensions Guide

Symbol	**Description**
N	Number of datapoints
p	Number of dimensions/attributes
O	Number of outputs per datapoint

Table 1.2: Notation Guide

Symbol	**Description**	**Dimensions**
$\mathbf{X}$	Input matrix	$N \times p$
x_i	Column vector of a row in $\mathbf{X}$	$p \times 1$
$\mathbf{X}_j$	Column vector of a column in $\mathbf{X}$	$N \times 1$
$\mathbf{y}$	Expected output matrix	$N \times O$
y_i	Row vector of a row in $\mathbf{y}$	$1 \times O$
$\hat{Y}$	Predicted output matrix	$N \times O$

1.1.2 Nearest-Neighbor Methods

CHAPTER 2

Glossary

Table 2.1: Useful terms, where they appear in the text, and potential synonyms

Term	**Page**	**Synonyms**
Bias	5	Intercept
Categorical	5	Discrete, Qualitative
Cost Function	TEMPORARY	Error Function, Loss Function
Classification	5	N/A
Discrete	5	Categorical, Qualitative
Error Function	TEMPORARY	Cost Function, Loss Function
Feature	5	N/A
Intercept	5	Bias
Learner	5	N/A
Loss Function	TEMPORARY	Cost Function, Error Function
Qualitative	5	Categorical, Discrete
Quantitative	5	N/A
Regression	5	N/A
Residual Sum of Squares	6	N/A
Test Set	5	N/A
Training Set	5	N/A
Validation Set	5	N/A
Vector of Coefficients	5	N/A