Exam 2021

June 21, 2023

Exercise 0.1. Ordinal regression refers to a type of supervised learning problem where (discrete) taget labels show a natural ordering. For example, classifying wines in a 1-10 scale, or predicting customer satisfaction into one of excellent, good, average, or bad. Given the type of supervised learning problem that you have come accross during our course, what modifications or extensions can you think of that could solve the ordinal regression problem? You can mention how to alter learning algorithms, or cost functions, or anything you can think of.

- Extend Existing Algorithms: Many popular classification methods, such as logistic regression, can be extended to handle ordinal data. For instance, Ordinal Logistic Regression (also called Ordered Logit model) can be used when the response variable is ordinal in nature.
- Treat as Regression: Another method would be to treat the problem as a regression problem. Since the labels are ordered, they can be treated as numerical data. However, caution should be taken since this assumes equal intervals between the classes which might not be the case in some datasets.
- Cost Functions: Design a cost function that takes the order of classes into account. For example, a misclassification from "good" to "bad" might be penalized more than a misclassification from "excellent" to "good". You may design a new loss function or modify an existing one to reflect this.
- Pairwise Comparison: You can transform the problem into binary classification tasks by comparing all pairs of classes.
- Cumulative Comparison: Another transformation option is to consider the problem as a set of binary classification tasks that discriminate between lower ranks and higher ranks.

Exercise 0.2. A data analyst has received a (small) set of expensive, labelled data and wants to build a good predictive classification model. She comes up with the following protocol:

1. Split all available data into training, validation and test sets using 50/25/25 proportions at random
2. Train SVM model with default parameters and polynomial kernel of degree 3 on the training set
3. Train and optimize Random Forest, optimizing its hyper-parameters using OOB on the training set
4. Choose best of SVM, and RF models using error on the validation set
5. Estimate true error of selected model using test set

Please criticize (in a constructive manner) this solution.

The proposed method could work well, but there are a few potential issues:

- Data Split: For a small dataset, it might be better to use a method like cross-validation rather than a static split. This can help the model see all the available data during training and validation phase.
- Parameter Tuning: SVM with a polynomial kernel of degree 3 is used with default parameters. It would be better to tune these parameters (like the C parameter for regularization) to ensure that the model is not underfitting or overfitting.
- Model Selection: Choosing the best model based solely on validation error might be a bit simplistic. Other metrics (like precision, recall, F1-score) could give a more rounded understanding of how the model performs.

- Hyperparameter Optimization: For Random Forest, optimizing hyperparameters using OOB error can be a good choice. However, it should be noted that hyperparameters should ideally also be validated using the validation set or cross-validation. The OOB error is a good rough estimate but it might not fully represent the model's ability to generalize.
- Model Bias: The protocol includes two algorithms only (SVM and RF). If the data characteristics favor other types of models (e.g. neural networks, gradient boosting machines etc.), they could be overlooked. Including a wider array of models for comparison might be beneficial.

Exercise 0.3. Explain in your own words the difference between the posterior and the predictive distribution in Bayesian learning.

In Bayesian learning, the posterior distribution and the predictive distribution play significant roles, but they serve different purposes:

- Posterior Distribution: The posterior distribution is the probability distribution of an unknown quantity, treated as a random variable, conditioned on the observed data. In Bayesian learning, it's used to represent our updated belief about the model parameters after we have seen the data. It combines our prior knowledge, expressed as the prior distribution, with the evidence provided by the observed data in the form of the likelihood function.
- Predictive Distribution: The predictive distribution is the probability distribution of a new, unobserved data point given the observed data. It integrates over all possible parameter values, each weighted by their posterior probability. In other words, it provides a distribution over possible values for new data points, taking into account parameter uncertainty.

Exercise 0.4. Please mark whether the following statements are true or false;

- Training error is always lower than test error: T (in general)
- Lasso and ridge regression both help in reducing overfitting: T
- Lasso regression is preferable to ridge regression because it produces sparse models: F (it produces sparse model, but this is not necessarily preferable)
- The activation functions of output neurons of a neural network are determined mainly by the nature of the target variable one wants to predict: T
- Linear regression assumes Gaussian input variables: F
- Naive Bayes assumes Gaussian input variables: F
- Bayes formula is used in Bayesian learning to obtain posterior distributions: T
- It is not possible to train a neural network for both regression and classification at the same time: F
- Bigger training sets help to reduce overfitting: T
- It is impossible to evaluate the quality of a clustering result because we never have the ground truth: F
- Cross-validation is a resampling method used to select a good model: T
- Gaussian Naive Bayes assumes Gaussian input variables: T
- Backprop is an algorithm used in neural network learning to obtain partial derivatives of an error function with respect to its weights: T
- The negative log-likelihood can always be used as an error function in supervised learning: F
- The EM algorithm is particularly suited to learn probabilistic models with partially observed data: T