Notes on Classification

Sameer Kesava PhD

Contents

Improving the Accuracy

A robust and accurate model can be achieved by tackling the **bias-variance** problem.

$$Bias = <\hat{f}(X)> - f(X) \tag{1.1}$$
$$Variance = <\hat{f}(X)^2> - <\hat{f}(X)>^2 \tag{1.2}$$
$$<[f(X) - \hat{f}(X)]^2> = Bias^2 + Variance \tag{1.3}$$

1.1 Datasets

- Train, valid and test datasets should contain similar ratio of number of classes.
- The above datasets must be disjoint, i.e. any redundancy must not be used.
- If there are missing values, it is safe to input such values as zero if the value does not carry any weight.
- If test data is expected to have missing values, ensure training data does too so tht the NN can learn to ignore such values; can be artificially generated as noise.

1.2 Weights

1. Sample or Class weights can be used to improve the prediction accuracy, e.g. assigning large penalties to wrong predictions of the minority class using the 'class weight' parameter.

2. Alternatively, a user-defined function can be supplied to compute the weights.

3. Weighted micro and macro-averaging useful for scoring imbalanced datasets.

1.2.1 ML

1.2.2 DL

(a) A good starting values for the initializers are Glorot-Xavier or Orthogonal for the weights.

(b) For the bias, zeros suffice.

1.3 Cross Validation

- Can also use ***Bootstrap***.
- Once the best model is chosen through CV, then the model should be trained on the whole dataset for the final model.

1. Use k-fold CV. k = 10 is a good starting value.
2. Use stratified k-fold to preserve class proportions.

1.4 Regularization

1.4.1 LASSO or L1-norm

A useful feature selection technique

1.4.2 Ridge Regression or L2-norm

L2 regularization also called ”weight-decay” in DL.

1.4.3 L1-L2 norm

- Implemented as ElasticNet in scikit-learn.

1.5 Brute-Force

Brute-force algorithm is effective for small datasets.
Query time grows as O[pn], where p is the number of features/predictors/dimensions and n is no. of data points.

1.6 Small Dataset

Also see Augmenting Data section 1.9. n: number of samples
p: number of features/predictors

If different shuffling rounds during training leads to very different metrics during training, then the size of the dataset might not be sufficient.

Careful feature engineering helps train a better model.

When p $>$ n, then there is no unique solution - variance is ∞.

1.6.1 When p is comparable to n

1. Use shrinkage methods such as Lasso or L1 norm with k-fold CV for feature selection.
2. Check Brute-Force section 1.5.

1.6.2 p $\ll$ n

1. Use k-fold CV. Also, iterated k-fold CV where the dataset is shuffled before k-fold splitting.

1.6.3 Batch Size

If feasible, training should be tried on the whole data at once rather than splitting into mini-batches.

1.7 Class Imbalance

Class imbalances can be handled using the following approaches.

- Upsampling.

- Downsampling.
- Generation of synthetic training samples using methods such as SMOTE.
- Data augmentation; see section 1.9.

1.8 Ensemble Learning

Random Forest, for example, is a type of ensemble learning method as different trees train on different subsets of the samples(?).

- Use Nested CV for ensemble learning.
- Boosting
- Bagging
- Stacking
- Majority/Plurality voting for multiclass setting.
- Cloning estimators (sklearn.base import clone) for fitting on different formats of training data.

1.8.1 Bagging

- Also called Bootstrap aggregation.
- Used in Random Forests, random feature subsets are selected with replacement.
- Improves accuracy of unstable models; at the expense of interpretability.
- Reduces overfitting by reducing variance.
- No. of trees is not a critical parameter; using large no. does not lead to overfitting when using bagging.
- Out-of-Bag MSE
  1. For quantifying test error without the need of CV.
  2. Convenient when working with large datasets as CV could be computationally onerous.

3. For sufficiently large B, OOB error is virtually equivalent to LOOCV error.

- Variable Importance measures for feature importance.
- Ineffective in reducing model bias. Hence, should be used in conjunction with ensemble classifiers with low bias such as unpruned decision trees, i.e. where max_depth = 0.

1.8.2 Boosting

- Does not involve bootstrap sampling.
- Boosting learns **slowly**. In general, statistical learning methods that learn slowly tend to perform well.
- Boosting with trees: no. of trees affects the extent of overfitting. Hyperparameters are
  1. No. of trees. Small no. aids in model interpretability.
  2. Shrinkage λ
  3. Interaction depth
- Adaptive Boosting (from sklearn.ensemble import AdaBoostClassifier)
  1. Focus on samples that are hard to classify.
  2. Can lead to decrease in both bias and variance.
  3. Uses an ensemble of classifiers: the i+1th learner has it's input the output of the ith learner.

1.8.3 Stacking

- For reducing bias.
- Uses different learners and then a meta-learner in the end which takes the output of different learners to yield the response.

1.9 Augmenting Data

Accuracy can be improved by augmenting to the data and then fitting the model with both the original and augmented datasets. Some examples are

- Adding random noise to the data.
- In case of images, random rotations of the images.
- Using too many augmented images is also not advisable as these images are inter-correlated.

1.10 Hyperparameter Tuning

- Hyperparameter optimization based on Gaussian Process Regression and Bayesian Optimization
- keras tuner in keras
- GridSearchCV or RandomSearchCV in scikit-learn
- RandomSearch can be used as the baseline against which optimization algorithms can be evaluated.

ML Methods

Attempt statistical ML methods before DL.

2.1 Bayes Classifier

Since the true distribution is not known, Bayes Classifier cannot be calculated.

2.2 Random Classifier

For comparison with ML models.

1. Random shuffling of a copy of test labels in an array or tensor.
2. Equating to the original, unshuffled copy.
3. The percentage of True values give

2.3 Naive Bayes Classifier

Highly useful for text classification, works well for relatively small datasets and computationally efficient.

2.4 Logistic Regression

- Can be used as the base model.
- The function is gives the conditional distribution of response Y, given predictors X.

- Quadratic logistic regression can also be tried.
- **GAMs** (see Regression document) can also be explored with logistic regression for multivariate data.
- It is closely related to Support Vector Machine section 2.7.
- When there are more than 2 classes, there are 2 approaches:
  1. One-vs-One
  2. One-vs-All

2.5 Linear Discriminant Analysis

LDA classifier assumes that the observations are drawn from classes with a normal distribution of predictors while sharing a common variance σ^2.

2.6 Quadratic Discriminant Analysis

QDA classifier assumes that the observations are drawn from classes with a normal distribution of predictors but not sharing a common variance σ^2.

2.7 Support Vector Machine

- Hyperparameter C controls the bias-variance tradeoff.
- Non-linear decision boundaries can be addressed by enlarging the feature space using higher order polynomial terms.
- Kernels
  1. Linear kernel: quantified using Pearson correlation.
  2. Kernels for non-linear decision boundaries: polynomial, radial.
- Very low sensitivity, like Logistic Regression, to observation far from the decision boundary.
- When classes are well-separated, SVMs tend to behave better than logistic regression, and in more overlapping regimes, logistic regression is often preferred.

2.8 Decision Tree

- Does not require any data transformation, i.e. scaling, before training.
- **Gini-impurity** and **entropy** are the measures of impurity.
- The splitting is based on a top-down, greedy approach. At each node, search is carried out for the best predictor and cut-off point by minimizing RSS in the case of regression.
- Tree pruning for reducing variance.
- Sometimes, the splits will yield terminal nodes that have the same predicted value. This is performed because it leads to node purity increase, which helps in the confidence in predictions.
- Sometimes, even if under-performing compared to linear models, trees are preferred for the sake of interpretability.

2.9 Random Forest

- Ensemble of decision trees.
- Random Forest with m = p is bagging trees, where m is the number of predictors out of p chosen for splitting.
- This is an improvement over bagging trees by way of a method that decorrelates the trees.
- Can use large no. of trees for the test error rate to settle down, i.e. test error vs no. of trees.
- OOB (see Bagging subsection 1.8.1) for test error rate.
- Tree-pruning for reducing variance.

2.10 XGBoost

Highly popular classifier and regressor.

- Gradient boosting method
- Absolute loss and Huber loss more robust to outliers.

2.11 Latent Dirichlet Allocation

- Unsupervised Learning
- For topic modeling, e.g. classifying a newspaper as a topic.
- scikit-learn lda module for batch and online learning.

DL Methods

All operations, e.g. loss functions, activation functions, in neural networks must be differentiable.

3.1 Binary Classification

- Should end with a dense layer of 1 unit and a sigmoid activation function.
- The loss function should be binary cross-entropy.

3.2 CNN

- Convolutional layers learn local patterns whereas dense layers learn global structure.
- In CNNs, the higher the layer is, the more specialized it is. The first few layers learn very simple and generic features which generalize to almost all types of images. The higher the layer is, the features learnt are increasingly specific to the dataset on which the model is trained.
- The layers are defined by 2 key parameters
  1. kernel size: typically 3x3 or 5x5
  2. Filters: 32, 64, typically of the order of 2^n
  3. Pooling operations are advised over strides for downsampling.
- Typically, as the layer number increases, the no. of feature maps increases and the matrix dimension decreases (downsampling using pooling operations).
- Invariance is hard-coded in CNNs.(?)

3.2.1 Transfer Learning

- When using pretrained models, use the convolutional base for feature extraction.
- The dense layers are largely useless for object detection.
- If the dataset differs a lot from the dataset used for the pretrained model, then not all CNN layers should be used, only the first few layers (a probable hyperparameter) are advised to be used.
- There are 2 ways to use a pretrained model (without fine-tuning)
  1. Use the base for feature extraction and its output as input for custom model. Apparently, augmented data cannot be used here. not sure why? (from F.Chollet DLP book)
  2. Use the base as part of the model but with its trainable parameters frozen and then train the model with additional custom layers built on top.

3.2.2 Spatial Pyramid Pooling Layer

For images of arbitrary dimensions (width and height).
Majority of transfer learning models require input data with square dimensions, i.e. width = height.

Loss Functions

Loss functions must be defined carefully based on the problem statement.

In multiclass datasets, if each data point belongs to a single class, then the problem is **"single label, multiclass classification"**. If it could belong to multiple categories. e.g. topics of newspaper articles, image classification, then the problem is **"multilabel, multiclass classification"**.

4.1 Metrics for Quantifying Test Errors

1. Adjusted R^2
2. C_p
3. AIC
4. BIC

4.2 Sparse Categorical Cross Entropy

To be used when the labels are not one-hot encoded.

4.3 Binary Cross Entropy

1. If the labels have only 2 unique values, preferably [0,1] or [-1,1].
2. Use sigmoid activation function for [0,1] and tanh for [-1,1].

4.4 Kullback-Leibler Divergence

If each label is a probability distribution, e.g. image segmentation.

4.5 Perplexity

Loss function = e^{Loss}

Optimizers

1. If the training and validation accuracies with epoch jump around, then learning rate must be reduced.

5.1 Stochastic Gradient Descent

- For training one sample at a time.
- For online learning.

5.2 Mini-batch Gradient Descent

- For mini-batches typically of the order of 2^x for GPU compatibility.
- Examples are Adam, Adagrad, RMSProp, etc.

5.3 Momentum

Helps address convergence speed and local minima.

5.3.1 Nesterov

Overfitting

- Overfitting is a result of over-optimization over generalization.
- Random perturbation of the input data, e.g. images, to improve generalization.

6.1 ML

- See Bagging subsection 1.8.1
- L1,L2 regularization

6.2 DL

- More data
- L1, L2, L1L2 regularization
- Dropout
- Batch Normalization
- Reduce the complexity of the network especially if the data size is small.
- Early stopping
- Add noise. See Augmenting Data section 1.9.

6.2.1 Dropout

- Dropout functions
- Typical rates: 0.2-0.5

1. Gaussian Dropout

Diagnostic Tools

To gauge the performance of a learning algorithm, i.e. if a model suffers from high variance or high bias.

- If the accuracies are lower than the desired accuracy, then the model has high bias. This can be resolved by increasing the complexity of the model.
- If val. acc. $<$ train. acc, then this could be resolved by
  1. Collecting more data
  2. Increasing regularization strength
  3. Feature selection (L1 norm or LASSO)
  4. Feature extraction (PCA, UMAP)
  5. Reducing model complexity

7.1 Confusion Matrix

Class specific performance is crucial in medicine, biology, finance, etc.
Metrics in sklearn.metrics module.

7.1.1 Precision

TP/(TP+FP)

7.1.2 Sensitivity or Recall

TP/P

7.1.3 Specificity

1 - FP/N

7.2 ROC Curve

AUC value close to 1 implies a good classifier.
Crucial when working with imbalanced classes.

7.3 F1-score

$$2\frac{PRE * REC}{PRE + REC}$$

7.4 Learning Curves

Plotting training and validation accuracies as a function of training data set size.
sklearn.model_selection.learning_curve

7.5 Validation Curves

Plotting validation losses/accuracies as a function of epochs/iterations.
sklearn.model_selection.validation_curve

7.6 Test Error vs p

- If there are large no. of perdictors, then plotting test error vs p should give an idea of the number of predictors.
- Test error typically increases with number of predictors if the additional predictors are not related to the response.
- In high p scenario, multicollinearity between predictors is extreme. See VIF in Data_Preprocessing.

7.7 MSE vs R^2

7.8 Custom Scorer

Custom scorer function can be used with sklearn.make_scorer.

Statistical Tests

8.1 Bootstrap

Sampling with replacements, used to quantify the uncertainty in the parameter estimates.

8.2 T-statistic

When you run a hypothesis test, you use the T statistic with a p value.

$$t - statistic = \frac{\hat{\beta}}{SE(\hat{\beta})} \tag{8.1}$$

$\hat{\beta}$ is the coefficient estimate from the fitting.

8.3 Z-statistic or Z-score

statisticshowto.com: Simply put, a z-score (also called a standard score) gives you an idea of how far from the mean a data point is. But more technically its a measure of how many standard deviations below or above the population mean a raw score is.
A z-score can be placed on a normal distribution curve. Z-scores range from -3 standard deviations (which would fall to the far left of the normal distribution curve) up to +3 standard deviations (which would fall to the far right of the normal distribution curve). In order to use a z-score, you need to know the mean and also the population standard deviation .
Z-scores are a way to compare results to a normal population. Results from tests or surveys have thousands of possible results and units; those results can often seem meaningless. For example, knowing that someones weight is

150 pounds might be good information, but if you want to compare it to the average persons weight, looking at a vast table of data can be overwhelming (especially if some weights are recorded in kilograms). A z-score can tell you where that persons weight is compared to the average populations mean weight.

8.4 Confidence Interval

For normally distributed error, 95% confidence interval corresponds to 2σ.

Definitions

9.1 Images

1. **Aliasing**: Images with object boundaries as jagged edges.
2. **Anti-Aliasing**: Smoothening the boundaries to remove the jagged edges.

9.2 Variables

1. **Latent Variable**: Variables which cannot be measured directly such as quality of life, business confidence, morale and happiness but can be inferred from measurements of observable variables such as wealth, employment, etc.

9.3 Classification

1. **Multiclass**: The label/target belongs to multiple classes, e.g. MNIST classification
2. **Multilabel**: Multiple labels in one target, e.g. the topics to which a newspaper article belongs, an image containing multiple objects (cats and dogs for example).