Notes for Machine Learning Seminar

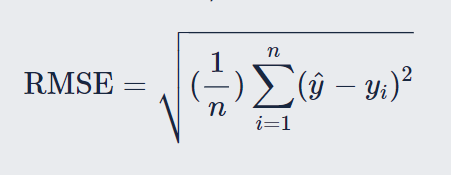
Slides: https://mlca.mm218.dev/2021

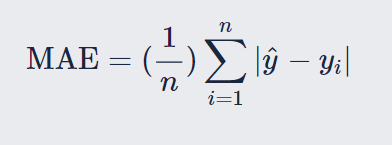
# 9/1/21 Prediction, Estimation, Attribution

* Read this: Bradley Efron: Prediction, Estimation, and Attribution.
* Differences in terminology from “regular” stats -> machine learning
  + Independent variable -> predictor
* Estimation vs attribution vs prediction
  + Estimation: aims to determine how some outcome varies over different levels of input variables
  + Attribution: Distinguish causal vs. correlated variables
  + Prediction: focus of this course, very different
* Basics of prediction
  + Prediction model only care about one thing: accuracy
  + Best model = one with fewest errors
  + Underlying mechanisms don’t matter if the predictions are accurate
  + Use root mean squared error (RMSE) is the most common fit metric (0 = perfect)
* Machine learning definition
  + any program which can use input data to make predictions
  + CAN include linear regression, but is usually it involves algorithms that don’t do estimation or attribution, and don’t worry about true relationships b/t variables
* Machine learning departures from traditional statistics (besides what is listed above)
  + Keep even weakly-correlated variables
  + Models don’t inform provable scientific truth

# 9/9/21 Regression

* Supervised learning
  + Some idea of the “true” values model “should” be predicting
  + Examples
    - Likelihood a species occurs at sites across the landscape
    - Extinction probability
    - Forest biomass
    - Annual temperature patterns into the future
* Models used to predict numeric outcomes are, by definition, regression models
* Multiple linear regression
  + Multidimensional planes are referred to as hyperplanes or surfaces
* Linear model assumptions
  + An underlying linear relationship (linearity)
  + Normality
  + Lack of multicollinearity
* Error in regression
  + Error formulas are a type “loss function” in machine learning context (sometimes called a “cost function”)
  + RMSE is one option (usually preferred)



* + Mean absolute error (MAE is another option)
    - States “model mispredicts X by an average of MAE”
    - Caution 1: MAE can also be “median absolute error)
    - Caution 2: MAE has a lower penalty for larger errors than RMSE
* Holdout (or simple) validation
  + Can remove some of the data to train model, then test on removed data
  + Datasets are called “training set” and “test set;” sometimes “holdout set” is used
  + Common training : test ratio is 80:20
  + Assumptions
    - Training data should resemble test data
    - Test data should resemble real-world data we care about predicting accurately
    - Test data should be completely unknown to model and modeler (model shouldn’t be tweaked based on test data)
  + How to “calibrate” model using “test” data in an acceptable way
    - Do a 60/20/20 split into training/validation/test dataset, use validation dataset to tweak model

# 9/15/21 Classification

* Classification models are used to predict categorical outcomes
* Can return numeric results; e.g., probabilities
* Focusing on binary classification this week (multiclass problems in two weeks)
* Linear models are not appropriate for classification, because they don’t predict probability
* Logistic models are better
  + Same assumptions as linear models
  + Confusion matrix provides better assessment tool than just straight predicted vs. observed (see R code)
    - Rows are prediction, columns are reference
    - Positive and negative predictive values are useful they’re (true positives)/(all positives) and (true negatives)/(all negatives), respectively; uses rows of prediction matrix as denominators
    - Specificity gives true negative rate (true negatives)/(true neg + false pos)
    - Sensitivity gives true positive rate (true pos)/(true pos + false neg)
    - Specificity and sensitivity use confusion matrix columns as denominators
    - Sensitivity and specificity are naturally opposed
    - Lots of other metrics can be calculated from confusion matrices (see Wikipedia page)
  + ROC curves
    - Abbreviation no longer means anything
    - Shows tradeoff between sensitivity and specificity
    - Often plotted as sensitivity vs FALSE positive rate
    - Top left corner is perfect accuracy
    - Allows user to set cutoffs for sensitivity and specificity; can use “best” point on curve (closest to top left corner) for a tradeoff between the two
    - Diagonal line (gray in R) shows a completely random model
    - Can use area under curve (AUC) to assess model
    - The most common rule-of-thumb for AUC says:
      * If AUC == 0.5, then our model is no better than flipping a coin
      * If 0.5 < AUC < 0.7, the model is a "poor" classifier
      * If 0.7 ≤ AUC < 0.8, the model is "acceptable"
      * If 0.8 ≤ AUC < 0.9, the model is "excellent"
      * If 0.9 ≤ AUC, the model is "outstanding"

# Final Project Notes from 9/15/21

* Basically a mini scientific paper
* Also turn in code w/ an Rproj file

# 9/22/21 Imbalanced Classification

* Imbalanced classification problem definition: problem with imbalanced classes, i.e., there are a lot fewer cases of one (or more) class(es) than others
* Ways to deal with this problem
  + Change probability threshold using ROC curve
  + Weight observations so classes are treated equally
  + Resample
    - “Super sample” smaller classes to artificially increase sample size
* Imbalanced classification can be used intentionally as a tool to tune model for sensitivity or specificity (e.g., credit card fraud; want lots of sensitivity)
* No perfect solution to imbalanced classification yet

# 9/29/21 Decision Trees

* Notes from in class
  + Types of error
    - Bias
    - Variance
    - Irreducible (can’t fix this)
  + Definition of modeling bias
    - Shape of model is not a good fit for shape of data
  + Bias vs variance tradeoff
    - High bias caused by rigid model; data should look like model
    - High variance: model made from data should look like the rest of the data
* Tradition parametric regressions have assumptions that are almost always broken
  + Assumption refresher
    - Normally distributed residuals
    - Linear associations between predictors and outcomes
    - Independence
    - Non-correlated predictors
  + Most commonly issues in natural sciences are lack of independence and correlated predictors
* Timeline of analysis methods
  + Linear regression: 1805
  + Logistic regression: 1838
  + Decision trees: 1960 (designed for social scientists)
  + Our method of decision tree: 1974
* Decision trees are rarely the best tool; here are alternatives:
  + Highly explainable non-linear model: general additive model
  + Maximize predictive accuracy: random forests and gradient boosting machines
    - Decision trees underpin more advanced ML techniques
* Description of a decision tree
* Concept of impurity
  + Decision trees create high bias, low variance models because they try to minimize impurity
  + Can only be used for classification (not regression; regression decision trees minimize variance in y-value)
* Decision trees are notorious for overfitting
* Misc: use ‘bench’ package to time code

**10/6/21 Random Forests**

* Motivation
  + Decision trees are high variance and overfit
  + Idea to reduce variance: fit decision trees to many training sets
  + Average of predictions could smooth hard thresholds
* Bootstrap aggregation (“bagging”)
  + Precursor to a random forest
  + Decision trees perform better on large datasets
  + Resampling allows for relatively large training datasets without dipping into the testing dataset
  + Sampling with replacement in this manner is bootstrapping; each sample is a bootstrap sample
  + Bootstrap aggregation: use mean prediction of bootstrapped decision trees to get a more accurate prediction
  + Decision trees are weak learners; combining weak learners can improve predictive accuracy
  + To further increase predictive power, you can use a random subset of predictors for each model
* Random forest
  + Extension of bootstrap aggregation
  + Instead of fitting each tree to a different set of variables, we chose a random group of variables for each split
  + Fits better than bagged tree models

# 10/13/21 Hyperparameter Tuning

* Cross validation
  + Conundrum: want to evaluate different hyperparameter tunings, but don’t want to invalidate the test set
  + Solution: cross validate by creating subsets of the data
  + K-fold cross validation (k-fold CV)
    - Divide training set into k number of folds, fit k number of models, excluding a different fold from each model

Table

Description automatically generated

* + - Leave-one-out cross validation (LOOCV) is the most extreme form of k-fold CV, where each observation is its own fold; commonly used to assess model accuracy where a holdout test set isn’t possible
    - 5 or 10-fold CV is more common LOOCV; results are generally similar enough between the types of k-fold CV
    - Some papers use cross-validation without a test set; we’re not doing this
    - Can’t do randomly assigned fold for everything; time series data is one example; “rolling forecast origin evaluation” is better textbook Forecasting: Principles and Practice (<https://otexts.com/fpp3/tscv.html>)
* Hyperparameter tuning for regression
  + Ideal tuning values varies a lot between R packages and ML methods
  + Grid search can be used to find good combination
    - Random grid search often better: make grid with all possible combos of hyperparameters
    - Note: if a hyperparameter is very variable within the top hyperparameter sets, its value likely doesn’t matter
  + For random forests, more trees are almost always better; good rule of thumb is 10 x (number of predictors)
  + Namedrops of other hyperparameter tuning methods:
    - Bayesian optimization
    - Simulated annealing
    - Genetic algorithms
* Classification and cross-entropy loss
  + Cross-entropy loss function is the typical loss function (RMSE doesn’t work)
  + It’s just a negative log function
  + Example when true answer is yes
  + Calculate cross-entropy loss by taking the negative logarithm of the probability assigned to the correct class. Graphical user interface, application, Excel

    Description automatically generated with medium confidence
  + Caveats with cross-entropy
    - NOT as good model statistic (fall back on sensitivity, specificity, AUC)
    - Increases with number of observations
    - Best used for model comparisons only
    - Not a silver bullet for imbalanced classification (class weighting might still be needed)
    - Make sure that if you’re trying to predict a “no,” swap your 0s and 1s

# 10/13/21 A bit of machine learning background

* Machine learning lacks a lot of the mathematical underpinnings of traditional statistics
* Best predictive ML model types
  + Random forest
  + Stochastic GBMS and stacked ensembles
  + Support vector machines
* Epilogue of Computer Age Statistical Inference book is interesting

# 10/20/21 Gradient boosting machines

* Overview
  + GBMs are another way of aggregating decision trees to reduce variance and increase predictive accuracy
  + Grows trees in a sequence and predict residuals of prior trees
  + Computationally intensive and hard to tune
  + Often very good at providing accurate predictions
  + Gradient descent is movement toward a minimum RMSE
* Boosting
  + Def: sequentially adding models to a base model to reduce errors
  + Steps
    1. Fit a base model
    2. Get residuals
    3. Fit new tree to residuals
    4. Add predictions to old predictions, applying a learning rate weight to residual predictions (we used 0.5 in our
    5. Fit new model
    6. Return to step 2 until done
  + Weak learners, such as decision stumps (one split) are often employed to reduce overfitting
* Gradient boosting machines
  + Automatically fits boosted trees
  + Issues
    - Often performs poorly without intensive tuning
    - VERY sensitive to hyperparameter values
    - Tuning must be done sequentially due to large computational needs
    - Worse at removing error due to variance than random forests

A black screen with white text

Description automatically generated with low confidence

# 10/27/21 Stochastic GBMs and Stacked Ensembles

* This week focusing on GBM extensions that are less computationally expensive and do a better job at reducing variance error
* Stochastic gradient boosting machines fit data to a different, random subsample of the data on each run
* R packages of note for stochastic GBMs
  + xgboost (oldest mainstream package)
  + lightgbm (faster than xgboost and still accurate)
  + catboost (slower and less accurate than lightGBM but more easily deals with categorical variables)
* Stacked ensembles
  + Combines predictions of strong learner models (“stacking”)
  + Can average model predictions or fit a model to the model predictions to make further predictions
  + Can even stack grid searches
  + Pros
    - Reflects uncertainty about which model is “best”
    - Can make predictions more robust on unknown data
* NOTE: The presentation for this week has useful links to model averaging papers

# 10/27/21 LightGBM

* Resources for lightgbm
  + Hyperparameters: <https://sites.google.com/view/lauraepp/parameters>
  + Tuning guide: https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html
* Timeline of issues
  + Oct 9, 2021: new version
  + Oct 12: issues noticed
  + October something: Removed from CRAN
* Had to use a Microsoft CRAN snapshot
* Now need to send “…” to params

# 10/27/21 Reproducibility in R

* Option 1: Use as little package code as possible
* Option 2: Freeze dependencies
  + Microsoft’s MRAN daily snapshots (meh)
  + ‘renv’ package
* Option 3: Freeze your computer using Docker with R (rocker)

# 11/3/21 k-nearest Neighbors and Kernel-based ML

* Not going to mathematically define kernel for the purpose of this class; think of them as “distance-based” methods
* k-nearest neighbors (KNN)
  + Rarely the best model and inefficient to compute
  + Predicts class of an observation by looking at the k closest “neighbors” in a coordinate system
  + Want an odd number (or a number not divisible by the number of classes)
  + Need to center-scale all variables because KNN doesn’t understand units
  + KNN always uses all predictors; dimension reduction like PCA can be helpful
* Note on center-scaling when using a test set
  + To PROPERLY center scale a test set:
    1. Calculate mean and SD of each variable in the TRAINING SET ONLY
    2. Apply mean and SD to
  + Failing to center scale properly and allowing the test set to influence the center scaling is called data leakage
* Tuning KNN
  + Can’t reliably use cross-entropy for tuning because cross-entropy will usually decrease as k increases
  + Need to use AUC of ROC for tuning

# 11/10/21 Support Vector Machines

* Kernel-based
* Pros:
  + Flexible
  + Robust to outliers
  + Resilient to overfitting
* Cons:
  + Slow to train, especially with datasets with many observations
  + Rarely the best model outside of niche use cases
* Separating hyperplanes
  + Linearly separable: data clusters can be separated by a straight line
  + “Linear separator” is a type of separating hyperplane
  + Decision boundary: prediction depends entirely on which side of a separating hyperplane it falls
  + Optimal separating hyperplane
    - Hyperplane that maximizes the distance between the closest points between two point clusters
    - Points used to draw optimal separating hyperplane are support vectors
* Hard margin classifiers
  + Hard margin classifiers are the simplest support vector machines
  + Draw a margin, dotted lines parallel to optimal hyperplanes
  + If no points are in the margin and no points are misclassified, we have a hard margin classifier
  + Weakness: prone to outliersChart, scatter chart

    Description automatically generated
* Soft margin classifier
  + *C* is a parameter that determines how many points are allowed in the margin (it’s not C = 1 means one point is allowed; it can be any positive number)
  + If C is infinite, the furthest points are used to create the margin
* Dimension casting in SVMs
  + If one class “rings” another, it might be possible to add another variable to make data separable in a higher dimension

Chart

Description automatically generated

* + Adding dimensions in other ways can allow approximation of any non-linear relationships
  + Kernel functions are used to cast data into higher dimensions
  + Radial basis function (RBF) kernel is the default starter kernel
* More support vector machine (SVM) stuff
  + Tends to do poorly when there are too many categorical variables
  + Center scaling required for data
  + Loss functions are difficult for classification problems because there are no probabilities involved; use overall model accuracy; resample to rebalance classes if needed
* Support model regression
  + Goal: Find a dimension where all points line up and you can do a simple linear regression
  + Overfitting avoided by setting hyperparameter epsilon, which controls the acceptable variance around the hyperplane