Biostat 202: Opportunities and Challenges of “Big Data”:

Machine Learning 4:
Cross-Validation, Ensembles, and Feature Importance

Aaron Wolfe Scheffler
Division of Biostatistics
Department of Epidemiology and Biostatistics
Review of last lecture

1. K-nearest neighbors

2. Train/validate/test

3. Classification trees

4. Support vector machines

5. Artificial neural networks
Outline

• Cross-validation
• Ensemble methods
  • Bagging: *Random Forests*
  • Boosting: *Adaboost*
  • Stacking
• Parameter tuning, continued
• Feature importance
Cross-validation
Cross-validation

- For complex models and ensemble, parameter tuning is almost always required
- What if you don’t have enough data to split 3 or even 2 ways for train/validate/test?
- If data is expensive, it is a big sacrifice to reserve a large proportion of data for validation and testing...

- Solution: cross-validation
Cross-validation

• Cross-validation
  • First step: data is split into $k$ subsets (folds) of equal size
  • Second step: each subset is used in turn for validation (evaluation of prediction) and the remainder for training

• This is called $k$-fold cross-validation (10-fold is common choice, also 5-fold for smaller datasets approx. <200 and even $N$-fold for very small datasets, called leave-one-out)
E.g. 5-fold cross-validation

1. Break up data into 5 “folds” of the same size

2. Set aside one fold for validation and use the rest to build model, repeat this process for each fold

- Estimate the predictive performance of our model on the combined validation data (i.e. the validation_1-5 sets)
Cross-validation

• Notice that at no time is the predictive performance based on data that was used to train the model – so there is no over-fitting

• Use of cross validation
  • Single model assessment: replaces train/test
  • Multiple model assessment (i.e. hyperparameter tuning): replaces train/validation – STILL NEED TO ASSESS FINAL MODEL ON A TEST SET!
Cross-validation - Orange

• The evaluation metrics are averaged across folds to yield an overall assessment of predictive performance.

• Often the subsets are stratified before the cross-validation is performed – e.g. same proportion of each class or group in each fold.
Ensemble Methods
Ensemble methods

- Train a bunch of predictive models and pool them to form a final prediction.

**Advantage:**
- improvement in predictive accuracy

**Disadvantages:**
- more computation
- it is difficult to understand an ensemble of classifiers

**Examples:** bagging, boosting, and stacking
Bagging

• Bagging = “Bootstrap Aggregating”

• Simplest way of combining predictions from a similar class of models

• Train models on slightly different versions of the data and combine via voting or averaging with each model receiving equal weight

• You can bag any model, most common is decision trees (e.g. random forest)
Bagging

• Basic idea of bagging:
  1. Re-sample several training sets of size n from the dataset with replacement (bootstrap sample)
  2. Build a model for each training set
  3. Combine the models predictions (majority vote)

• Improves performance in almost all cases if learning scheme is *unstable* (e.g. decision trees depend heavily on where the first cut occurs)
Bagging

Original Data

Bootstrapping

Aggregating

Bagging

Classifier

Classifier

Classifier

Ensemble classifier
**Random Forests**

Training Data

Sampling with Replacement (bagging)

Bagged Sample  Bagged Sample  Bagged Sample

Training of Decision Trees w/ attribute bagging

Random Forest

**FINAL PREDICTION BY VOTING ACROSS ALL TREES, THE RANDOM FOREST!**
Boosting

• Use a sequence of “weak” learners to produce a strong one
• Uses model voting/averaging to aggregate predictions but models are trained on the same data and re-weighted based on their performance
• Justification: a sequence of weak learners that complement each other
Boosting

• Basic idea of boosting: iterative procedure:
  1. Train a weak model on the data
  2. Determine which observations are incorrectly/correctly predicted
  3. Retrain the weak learner, this time encourage it to try and guess the previously misclassified data points
  4. Repeat process (say 100 times) then vote across all models

• With each iteration, models are encouraged to become expert for instances classified incorrectly by earlier models

• Subtlety comes from in how misclassified instances are reweighted
Boosting
Adaptive Boosting (Adaboost)
Boosting algorithms

1997
Adaboost – Orange implemented

2016
XGBoost – considered best predictive algorithm for tabular data
## Boosting vs Bagging

<table>
<thead>
<tr>
<th>Single Classifier</th>
<th>Bagging</th>
<th>Boosting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single Iteration</strong></td>
<td><strong>Parallel</strong></td>
<td><strong>Sequential</strong></td>
</tr>
<tr>
<td><img src="image" alt="Single Classifier" /></td>
<td><img src="image" alt="Bagging" /></td>
<td><img src="image" alt="Boosting" /></td>
</tr>
</tbody>
</table>
Stacking

• Uses a “meta-learner” instead of voting to combine predictions across models
• Individual initial models can be of many different types
• Complex methodology to build meta-learner by learning again from the initial set of models (machine learning on top of machine learning)
• Can help unlearn bias in each base model
Stacking

- Initial dataset
- $L$ weak learners (that can be non-homogeneous)
- Meta-model (trained to output predictions based on weak learners predictions)
Parameter Tuning
Parameter tuning, continued

• Some algorithms have one parameter to tune
  • KNN: k neighbors
  • Linear/Logistic regression: alpha regularization parameter

• Others have many
  • Decision trees: number of splits, number of leaves, minimum number of subject, etc...
  • Random forest: everything about the tree plus number of trees

• Parameter tuning often requires extensive training and evaluation over a grid of possible hyperparameters
Parameter tuning, continued

• Example: decision tree
  • Min. number of instances in leaves [5, 10, 15]
  • Do not split subsets smaller than [15, 20, 25]
  • Limit the maximal tree depth to [20, 60, 80]
  • Produces 3x3x3=27 possible configurations!

• Parameter tuning is often done in an automated way by defining the parameter grid and telling the computer to find the configuration that produces a single optimal evaluation metric
Feature Importance
Feature importance

• Even though we fool ourselves into only being concerned with prediction, we often want to understand which features impact our prediction

• Given a target variable, which variables are important for prediction?

• The define this, you can develop measures of feature importance

• Feature importance can either be pair wise or model based
  
  • **Pairwise**: based directly on measures of association between target and individual features (*e.g.* correlation)
  
  • **Model based**: based on machine learning algorithms (*e.g.* regression coefficients, classification tree splits)
Feature importance – pairwise

• Looking at pairwise association between individual features and target can help identify variables that may be important for subsequent prediction

• Measures in Orange (target type dependent)
  • **Information gain**: expected information (reduction of entropy or chaos) in a feature with respect to a target
  • **ANOVA**: difference between average values of the feature in different classes
  • **Chi2**: dependence between features and outcome class
  • **FCBF**: information based measure, identifies redundancy due to correlation among features when estimating relevance to target
Titanic feature importance – pairwise

<table>
<thead>
<tr>
<th>Attribute</th>
<th>#</th>
<th>Info. gain</th>
<th>ANOVA</th>
<th>$\chi^2$</th>
<th>FCBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>2</td>
<td>0.218</td>
<td>NA</td>
<td>92.702</td>
<td>0.298</td>
</tr>
<tr>
<td>Fare</td>
<td></td>
<td>0.067</td>
<td>NA</td>
<td>66.603</td>
<td>0.000</td>
</tr>
<tr>
<td>Pclass</td>
<td>3</td>
<td>0.084</td>
<td>NA</td>
<td>54.466</td>
<td>0.075</td>
</tr>
<tr>
<td>Parch</td>
<td></td>
<td>0.019</td>
<td>NA</td>
<td>14.273</td>
<td>0.000</td>
</tr>
<tr>
<td>Embarked</td>
<td>3</td>
<td>0.021</td>
<td>NA</td>
<td>10.414</td>
<td>0.000</td>
</tr>
<tr>
<td>Age</td>
<td></td>
<td>0.005</td>
<td>NA</td>
<td>0.608</td>
<td>0.000</td>
</tr>
<tr>
<td>SibSp</td>
<td></td>
<td>0.029</td>
<td>NA</td>
<td>0.455</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Feature importance – model based

• Model output (if available)
  • Decision trees, regression
  • Rule induction (lab)

• Permutation feature importance – model agnostic
  1. Fit model with data and evaluate predictive performance
  2. Return to data, scramble one feature randomly among observations
  3. Refit model and compare model performance to original data.
  4. Repeat for every variable. Most important variables will produce biggest drops in predictive performance

• You can do this with any model that has tabular data!
<table>
<thead>
<tr>
<th>Height at age 20 (cm)</th>
<th>Height at age 10 (cm)</th>
<th>...</th>
<th>Socks</th>
</tr>
</thead>
<tbody>
<tr>
<td>182</td>
<td>155</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>147</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>142</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>130</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Feature importance – permutation
Titanic feature importance – model

The diagram shows a flowchart with nodes connected by arrows, indicating the flow of data through different processes. The nodes include 'Data', 'Random Forest', and 'Logistic Regression'. The flowchart is part of a larger system that processes data, possibly for machine learning or statistical analysis.

On the right side of the diagram, there is a table titled 'Rank' with columns for Scoring Methods, #, Info_gain, ANOVA, $\chi^2$, FCBF, Rand_rest, and Logis. The table ranks features based on various scoring methods, with 'Sex' having the highest Info_gain score of 0.218. Features like 'Pclass', 'Fare', and 'Embarked' also appear with their respective scores.

The table includes options to select attributes, with 'Best ranked' set to 5. The interface also notes that missing values will be imputed as needed.
Review of this lecture

• Cross-validation
• Ensemble methods
  • Bagging: *Random Forests*
  • Boosting: *Adaboost*
  • Stacking
• Hyperparameter tuning continued
• Feature importance
Next time

• Clustering
  • k-means clustering
  • TwoStep clustering
  • Kohonen self-organizing map (SOM)

• Data reduction
  • Principal components analysis (PCA)
  • t-Distributed Stochastic Neighbor Embedding (t-SNE)

• Guidance for choosing machine learning algorithms