Data Mining with Weka

Ensemble learning

Ian H. Witten

Ensemble learning

Committee structure: build different "experts," let them vote

- Often improves predictive performance
- Produces output that is hard to analyze
  - *but: there are approaches that aim to produce a single comprehensible structure*
- Methods
  - *Bagging*
  - *Randomization*
  - *Boosting*
  - *Stacking*

Ensemble learning

Bagging

- Several training sets of the same size
  - *produce them by sampling … with replacement*
- Build model for each one
  - *use same machine learning scheme*
- Combine predictions by voting
  (or, for regression, averaging)
- Very suitable for "unstable" learning schemes
  - *small change in training data can make big change in model*
  - *example: decision trees … but not Naïve Bayes or instance-based learning*
- Weka: meta>Bagging
- E.g. with glass.arff
  - *J48* *66.8%*
  - *Bagging (default parameters)* *72.4%*

Ensemble learning

Randomization: random forests

- Randomize the algorithm, not the training data
  - *how you randomize depends on the algorithm*
- Random forests
  - *attribute selection for J48 decision tree: don't pick the best, pick randomly from the k best options*
  - *generally improves decision trees*
- Weka: trees>RandomForests
  - *options: number of trees (default 10); maximum depth of trees; number of attributes*
- E.g. with glass.arff
  - *J48* — *66.8%*
  - *RandomForests (default parameters)* — *75.2%*

Ensemble learning

Boosting

- Iterative: new models are influenced by performance of previously built ones
  - *extra weight for instances that are misclassified ("hard" ones)*
  - *encourage new model to become an "expert" for instances misclassified by earlier models*
  - *Intuitive justification: committee members should complement each other's expertise*
- Uses voting (or, for regression, averaging)
  - *but weights models according to their performance*
- Often dramatically improves performance
- Weka: meta>AdaBoostM1
- E.g. with glass.arff
  - *J48* *66.8%*
  - *AdaBoostM1 (using J48)* *74.3%*

Ensemble learning

Stacking

- Combine predictions of base learners using a *meta learner* (not voting)
  - *base learners: level-0 models*
  - *meta learner: level-1 model*
  - *predictions of base learners are input to meta learner*
- Base learners are usually different schemes
- Can't use predictions on training data to generate data for level-1 model!
  - *Instead use cross-validation-like scheme*
- Weka: meta>Stacking
  - *and StackingC, more efficient version*
  - *allow multiple level-0 models (by specifying a metaclassifier)*
- Quite hard to make stacking work well, but with glass.arff I got
  - *J48* — *66.8%*
  - *StackingC: default metaclassifier + base classifiers IBk, PART, J48* — *72.5%*

Ensemble learning

- Combining multiple models into "ensembles"
  - analogy with committees of humans
- Diversity helps, especially with "unstable" learners
  - when small changes in the training data can produce large changes in the learned model
- Create diversity by
  - Bagging: resampling the training set — meta>Bagging
  - Random forests: alternative branches in decision trees — trees>RandomForests
  - Boosting: focus on where the existing model makes errors — meta>AdaBoostM1
  - Stacking: combine results using another learner (instead of voting) — meta>Stacking