Machine Learning Issues

Yeganeh Jalalipour

Practical Issues In ML

- Sample size
- Evaluation
- Overfitting
- Linearity
- Bad Data
- Feature Selection

Sample Size

- Induction with only a few samples is a fool's errand
- How much is enough?
- Worse, some of the samples need to be held out for evaluation. Tradeoff: more training samples = better accuracy (probably) but poorer validation

Evaluation

- Imagine training with *all* instances and then evaluating performance against all instances
- Brute force learner would be *perfect*
- Need to measure *generalization* across as-yet-unknown instances
- Typical method: hold out an *evaluation set*
  - ugh, less data for training
  - What if we are unlucky in our choice of evaluation set? Maybe training and evaluation set are not comparable anymore?

Cross-Validation

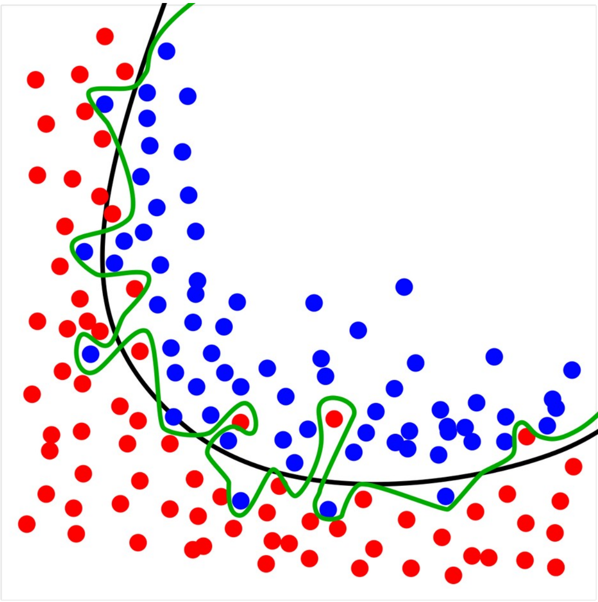
- Idea: Partition the data *S* into *n* equal subsets
- For each subset *S[i]* train on *S* - *S[i]* and evaluate on *S[i]*
- Do statistics on these *n* runs to get some kind of min/max/average accuracy
- Limiting case: "Leave-one-out" Cross-Validation; let $n = |S|$
- Cross-Validation is $n\times$ as expensive

Measures Of Accuracy

- For our binary case
- Name 0 0 true negative 1 1 true positive 0 1 false negative 1 0 false positive
- Once we have counted each of these, we can form various sums and ratios depending on what we want to do
  - Accuracy: $(tn+tp)/|S|$
  - Precision: $tp/(tp+fp)$
  - Recall: $tp/(tp+fn)$
- https://towardsdatascience.com/precision-vs-recall-386cf9f89488

Overfitting

- Never enough data
- Learner "masters" the training set, building a model that predicts it quite accurately
  - This mastery includes all the anomalies of the data set; outliers, and over-represented features
  - This degree of accuracy may *reduce* generalization, making the predictor *worse* on new instances



https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Overfitting.svg/1200px-Overfitting.svg.png

Controlling Overfitting

- Decrease amount of data in training set
- Have some principled measure of fit (Naive Bayes, Decision Trees)
- Use a validation set. Hold out more of the data and train on the training set until the performance on the validation set starts to get worse

Linearity

- Think of the feature vector as residing in an n-dimensional space
- A "linear" learner can find an n-1 dimensional plane in that space that best separates positive and negative training instances
- A "nonlinear" learner can find more complicated boundaries
- Linear: Naive Bayes, Perceptron
- Nonlinear: Decision Trees, k-Nearest Neighbor

Bad Data

- Real-world training instances will have:
  - Wrong classification
  - Mis-measured features
  - Missing features
- Algorithms need to be able to cope with this

Feature Selection

- Rare for a real-world inductive ML problem to come with instances that have a vector of Boolean features
- Choosing the right features makes a huge difference
  - Summarize the information useful for classification
  - *Leave out* features that can confuse the learner or kill performance
    - Consider a "random feature" that is computed for each instance by flipping a coin
    - This feature will be *accidentally correlated* with classification on small datasets, so learner will try to use it
    - It won't generalize well at all

Feature Types

- Boolean features allow all algorithms, but may lose information
- Set-valued features are only OK with some algorithms, require more data to exploit (hypothesis-space size)
- Scalar features only work with a few algorithms, but provide a lot of information (sometimes)
- Can always Booleanize a feature
  - Characteristic vector for set values
  - Scalar above/below mean, median
  - Scalar by gain split point