Statistical Techniques in Robotics (16-831, F10) Lecture #28 (November 30, 2010)

How to apply Machine Learning

Lecturer: Drew Bagnell *Scribe: Scott Satkin*

1. Plot your data
   - Histogram each feature
   - Scatter plot 2D/3D pairs or triplets of data
   - Perform PCA and scatter plot projected data
   - Unbalanced classes
     (a) Weight loss function or gradient
     (b) Upsample (duplicate data)
         – Downsides: overfitting and computational overhead
     (c) Downsample ("costing")
         – Downside: doesn't use all the data
2. Your hold-out data is sacred
   - Split data into training, validation and test sets. Be cautious of trajectory data (sequential information) or groupable data (partial exchangability, *e.g.*: different scenes or locations).
3. Cris Dima's rule:
   - "Always start by overfitting." (Get an upper bound on performance)
4. More data is better
5. Understand what features matter
   - Too many features
     – Computational problems
     – Overfitting problems
   - Ablative analysis
     – Remove features one at a time and see if performance drops and iterate.
     – Backwards or forwards greedy (*e.g.*: Boosting)
     – Analysis:
       * Greedy: $F_1 G_\infty (\log d)\sqrt{T}$
         Not good for sparse features.
       * Alternative: $F_2 G_2 \sqrt{T}$
         $\sum |w_i| \leq F_1 \subset F_2$
6. Never underestimate the power of a linear predictor (Linear SVM, Logistic Regression, Linear Regression...).

7. Never underestimate the power of "well tuned" Gradient Descent. (Nelder-Mead if no derivatives).
    - Normalize features $[-1, 1]$
    - Normalize standard deviations
    - Normalize each feature vector
    - Randomize data-points
8. Understand Overfitting vs. Underfitting
    - Overfitting
        - Test on development set
        - Regularize more
        - Feature selection
        - Get more data
    - Underfitting
        - More features
        - More sophisticated classifier
        - Less regularization
        - Kernelize or boost
        - Optimize better
9. Don't buy the hype!
    - Rich Caruana[1] papers
    - SVM
    - Random forests ("Random Kitchen Sinks")
    - ANN (often 2nd best option)
10. Building Large Learning Systems
    - Premature statistical optimization (spend time on the parts that matter)
    - Unit test learning code

[1] `http://www.cs.cornell.edu/~caruana/`