Parting Thoughts on Machine Learning

Ron Parr
CPS 271

Themes in Machine Learning I

• Model (all of) the data
  – HMM learning, Bayes net learning, linear discriminant analysis, Gaussian mixture models and k-means, naïve Bayes

• Model the predicted variable
  – Bayesian linear regression, logistic regression, naïve Bayes

• Minimize error on training set
  – Regression, PCA, SVMs, perceptrons, neural networks

• How different are these?
Themes in Machine Learning II

- Regularization
  - Prevent overfitting by penalizing “complex” solutions

- Computational learning and structural risk
  - Classifiers with more “freedom” require more data
  - SVMs as structural risk minimizers

- Priors
  - Use prior knowledge to favor certain solutions in cases of insufficient data

- How different are these ideas?

Themes in Machine Learning III

- Algorithms often drive machine learning research (and distinguish it from statistics), but

- The algorithm and the underlying optimization should always be kept distinct in one’s mind

- The underlying optimization and the underlying probabilistic model (if defined) should also be kept distinct
Trends in Machine Learning

• Move towards probabilistic/Bayesian interpretations
• Use of fancier optimization techniques
• Kernelization
• Multi-task learning
• Sparsification and feature selection (e.g., L1 regularization)
• Dimensionality reduction via manifold learning

How Does Reinforcement Learning Fit?

• Reinforcement learning is, in some ways, the platypus of machine learning

• Arguably the least successful and most important area of machine learning
  – Least successful? No empires built on RL yet.
  – Most important? RL makes decisions; without decision making, machine learning is disconnected from the rest of intelligence.
Cool Stuff We Didn’t Have Time For

Variational Methods

- Variational methods are a general family of methods that can be used to approximation functions – useful to, but not specific to machine learning
- Basic idea: Approximate a nasty function (distribution) with a nicer one from a parameterized family, but
  - We pose an optimization problem to find the closest “nice” function to the nasty one
  - We choose the nice function so that it provides a bound on the nasty one
- Simple example: Factorization
  - Approximate: $P(ABC)$ with $F(A)G(BC)$
  - Pick $F$ and $G$ in a clever way to be close to and bound (typically a lower bound in a case like this) $P(ABC)$
- Comments:
  - Powerful technique if used with sophistication
  - Can be used to provide both upper and lower bounds
  - Can replace nasty inference problems with nasty optimization problems
  - Adding the word “variational” doesn’t make a sloppy approximation a better one, but it might make your paper sound deeper
Semi-Supervised Learning

• Suppose you have access to a huge body of data, but only a small set of these data are labeled (e.g., images)

• Q: Can the unlabeled data be helpful in coming up with a good classifier?

• A: In many cases, yes!

Active Learning

• Supervised learning assumes all training data have labels

• Active learning requires the learner to ask for labels

• Useful model in cases where data are plentiful, but obtaining labels can be expensive
  – Landmine detection
  – Protein structure

• As with semi-supervised learning, this often involves some sort of modeling of the unlabeled data
Active Feature Acquisition

- Generalization of active learning
- Suppose we need to ask for both labels and features
- Applications: Scientific data where acquiring labels or features requires costly lab work

- Can be unified with active learning. See, e.g., work from Larry Carin’s group.

Multi-task Learning

- Multi-task learning seeks to develop a general approach to learning that can exploit shared structure between tasks
- Suppose you have learned how to bake cakes
- Start from scratch when learning how to bake muffins?

- Some issues with multi-task learning:
  - Is this just regular learning where the problems are drawn from a larger bag that includes several related problems?
  - What distinguishes multi-task learning? The problem formulation? The evaluation method? The solution technique?
Manifold Learning

• Data often live in a lower dimensional space that is embedded in a higher dimensional space
• Discovering the manifold on which the data reside may simplify learning because distances are measured more naturally
• PCA can’t “unroll” because it is a linear method

"Swiss Roll" example from Saul et al.

Final Thoughts about ML

• Machine learning is a field of vast practical and economic importance
• It is built upon fairly basic principles: Model the data and/or model the classifier and/or model reconstruct the training data (and these are all closely related)
• To use machine learning fruitfully:
  – Think clearly about your features
  – 1. Understand your assumptions/model
  – 2. Understand your optimization problem
  – 3. Understand your algorithm
  – Understand that the above 3 are different things
• Don’t be frightened by big math; it’s just a tool to accomplish 1-3 above.