CS 478 - Tools for Machine Learning and Data Mining

Introduction to Metalearning

November 11, 2013

The Shoemaker's Children Syndrome

- Everyone is using Machine Learning!
  - Everyone, that is ...
  - Except ML researchers!
- Applied machine learning is guided mostly by hunches, anecdotal evidence, and individual experience
- If that is sub-optimal for our "customers," is it not also sub-optimal for us?
- Shouldn't we look to the data our applications generate to gain better insight into how to do machine learning?
- If we are not quack doctors, but truly believe in our medicine, then the answer should be a resounding YES!

A Working Definition of Metalearning

- We call *metadata* the type of data that may be viewed as being generated through the application of machine learning
- We call *metalearning* the use of machine learning techniques to build models from metadata
- Hence, metalearning is concerned with accumulating experience on the performance of multiple applications of a learning system
- Here, we will be particularly interested in the important problem of metalearning for algorithm selection

Theoretical and Practical Considerations

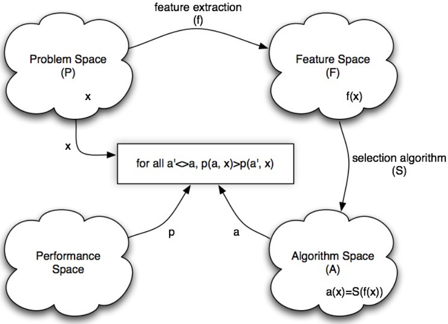
- No Free Lunch (NFL) theorem / Law of Conservation for Generalization Performance (LCG)
- Large number of learning algorithms, with comparatively little insight gained in their individual applicability
- Users are faced with a plethora of algorithms, and without some kind of assistance, algorithm selection can turn into a serious road-block for those who wish to access the technology more directly and cost-effectively
- End-users often lack not only the expertise necessary to select a suitable algorithm, but also the availability of many algorithms to proceed on a trial-and-error basis
- And even then, trying all possible options is impractical, and choosing the option that "appears" most promising is likely to yield a sub-optimal solution

DM Packages

- Commercial DM packages consist of collections of algorithms wrapped in a user-friendly graphical interface
  - Facilitate access to algorithms, but generally offer no real decision support to non-expert end-users
- Need an informed search process to reduce the amount of experimentation while avoiding the pitfalls of local optima
- Informed search requires metaknowledge
- Metalearning offers a robust mechanism to build metaknowledge about algorithm selection in classification
- In a very practical way, metalearning contributes to the successful use of Data Mining tools outside the research arena, in industry, commerce, and government

Rice's Framework

- A problem x in problem space P is mapped via some feature extraction process to $f(x)$ in some feature space F, and the selection algorithm S maps $f(x)$ to some algorithm a in algorithm space A, so that some selected performance measure (e.g., accuracy), p, of a on x is optimal



Framework Issues

- The following issues have to be addressed
  1. The choice of f,
  2. The choice of S, and
  3. The choice of p.
- A is a set of base-level learning algorithms and S is itself also a learning algorithm
- Making S a learning algorithm, i.e., using metalearning, has further important practical implications about:
  1. The construction of the training metadata set, i.e., problems in P that feed into F through the characterization function f,
  2. The content of A,
  3. The computational cost of f and S, and
  4. The form of the output of S

Choosing Base-level Learners

- No learner is universal
- Each learner has its own area of expertise, i.e., the set of learning tasks on which it performs well
- Select base learners with complementary areas of expertise
    - The more varied the biases, the greater the coverage
    - Seek the smallest set of learners that is most likely to ensure a reasonable coverage

Nature of Training Metadata

- Challenge:
  - Training data at metalevel = data about base-level learning problems or tasks
- Number of accessible, documented, real-world classification tasks is small
- Two alternatives:
  - Augmenting training set through systematic generation of synthetic base-level tasks
  - View the algorithm selection task as inherently incremental and treat it as such

Meta-examples

- Meta-examples are of the form $< f(x), t(x) >$, where $t(x)$ represents some target value for x
- By definition, $t(x)$ is predicated upon p, and the choice of the form of the output of S
- Focusing on the case of selection of 1 of n:

$$t(x) = \text{argmax}_{a \in A} \; p(a, x)$$

- Metalearning takes $\{< f(x), t((x) >: x \in P' \subseteq P\}$ as a training set and induces a metamodel that, for each new problem, predicts the algorithm from A that will perform best
- Constructing meta-examples is computationally intensive

Choosing f

- As in any learning task, the characterization of the examples plays a crucial role in enabling learning
- Features must have some predictive power
- Four main classes of characterization:
    - Statistical and information-theoretic
    - Model-based
    - Landmarking

Statistical and Information-theoretic Characterization

- Extract a number of statistical and information-theoretic measures from the labeled base-level training set
- Typical measures include number of features, number of classes, ratio of examples to features, degree of correlation between features and target, class-conditional entropy, skewness, kurtosis, and signal to noise ratio
- Assumption: learning algorithms are sensitive to the underlying structure of the data on which they operate, so that one may hope that it may be possible to map structures to algorithms
- Empirical results do seem to confirm this intuition

Model-based Characterization

- Exploit properties of a hypothesis induced on problem x as an indirect form of characterization of x
- Advantages:
  1. Dataset is summarized into a data structure that can embed the complexity and performance of the induced hypothesis, and thus is not limited to the example distribution
  2. Resulting representation can serve as a basis to explain the reasons behind the performance of the learning algorithm
- To date, only decision trees have been considered, where $f(x)$ consists of either the tree itself, if the metalearning algorithm can manipulate it directly, or properties extracted from the tree, such as nodes per feature, maximum tree depth, shape, and tree imbalance

Landmarking (I)

- Each learner has an area of expertise, i.e., a class of tasks on which it performs particularly well, under a reasonable measure of performance
- Basic idea of the landmarking approach:
  - Performance of a learner on a task uncovers information about the nature of the task
  - A task can be described by the collection of areas of expertise to which it belongs
- A *landmark learner*, or simply a *landmarker*, a learning mechanism whose performance is used to describe a task
- Landmarking is the use of these learners to locate the task in the *expertise space*, the space of all areas of expertise

Landmarking (II)

- The *prima facie* advantage of landmarking resides in its simplicity: learners are used to signpost learners
- Need efficient landmarkers
    - Use naive learning algorithms (e.g., OneR, Naive Bayes) or "scaled-down" versions of more complex algorithms (e.g., DecisionStump)
- Results with landmarking have been promising

Computational Cost

- Necessary price to pay to be able to perform algorithm selection learning at the metalevel
- To be justifiable, the cost of computing $f(x)$ should be significantly lower than the cost of computing $t(x)$
- The larger the set A and the more computationally intensive the algorithms in A, the more likely it is that the above condition holds
- In all implementations of the aforementioned characterization approaches, that condition has been satisfied
- Cost of induction vs. cost of prediction (batch vs. incremental)

Selecting on Accuracy

- Predictive accuracy has become the *de facto* criterion, or performance measure
- Bias largely justified by:
  - NFL theorem: good performance on a given set of problems cannot be taken as guarantee of good performance on applications outside of that set
  - Impossibility of forecasting: cannot know how accurate a hypothesis will be until that hypothesis has been induced by the selected learning model and tested on unseen data
  - Quantifiability: not subjective, induces a total order on the set of all hypotheses, and straightforward, through experimentation, to find which of a number of available models produces the most accurate hypothesis

Selecting on Other Criteria

- Other performance measures:
  - Expressiveness
  - Compactness
  - Computational complexity
  - Comprehensibility
  - Etc.
- These could be handled in isolation or in combination to build multi-criteria performance measures
- To the best of our knowledge, only computational complexity, as measured by training time, has been considered in tandem with predictive accuracy

Selection vs. Ranking

- Standard: single algorithm selected among n algorithms
  - For every new problem, metamodel returns one learning algorithm that it predicts will perform best on that problem
- Alternative: ranking of n algorithm
  - For every new problem, metamodel returns set $A_r \subseteq A$ of algorithms ranked by decreasing performance

Advantages of Ranking

- Ranking reduces brittleness
- Assume that the algorithm predicted best for some new classification problem results in what appears to be a poor performance
  - In the single-model prediction approach, the user has no further information as to what other model to try
  - In the ranking approach, the user may try the second best, third best, and so on, in an attempt to improve performance
- Empirical evidence suggests that the best algorithm is generally within the top three in the rankings

Metalearning-inspired Systems

- Although a valid intellectual challenge in its own right, metalearning finds its real raison d'être in the practical support it offers Data Mining practitioners
- Some promising implementations:
  - MininMart
  - Data Mining Advisor
  - METALA
  - Intelligent Discovery Assistant
- Mostly prototypes, work in progress: characterization, multi-criteria performance measures, incremental systems
- ExperimentDB