Active Learning with Multiple Views

Ion Muslea
SRI International, USA

INTRODUCTION

Inductive learning algorithms typically use a set of labeled examples to learn class descriptions for a set of user-specified concepts of interest. In practice, labeling the training examples is a tedious, time consuming, error-prone process. Furthermore, in some applications, the labeling of each example also may be extremely expensive (e.g., it may require running costly laboratory tests). In order to reduce the number of labeled examples that are required for learning the concepts of interest, researchers proposed a variety of methods, such as active learning, semi-supervised learning, and meta-learning.

This article presents recent advances in reducing the need for labeled data in multi-view learning tasks; that is, in domains in which there are several disjoint subsets of features (views), each of which is sufficient to learn the target concepts. For instance, as described in Blum and Mitchell (1998), one can classify segments of televised broadcast based either on the video or on the audio information; or one can classify Web pages based on the words that appear either in the pages or in the hyperlinks pointing to them. In summary, this article focuses on using multiple views for active learning and improving multi-view active learners by using semi-supervised- and meta-learning.

BACKGROUND

Active, Semi-Supervised, and Multi-view Learning

Most of the research on multi-view learning focuses on semi-supervised learning techniques (Collins & Singer, 1999, Pierce & Cardie, 2001) (i.e., learning concepts from a few labeled and many unlabeled examples). By themselves, the unlabeled examples do not provide any direct information about the concepts to be learned. However, as shown by Nigam, et al. (2000) and Raskutti, et al. (2002), their distribution can be used to boost the accuracy of a classifier learned from the few labeled examples.

Intuitively, semi-supervised, multi-view algorithms proceed as follows: first, they use the small labeled training set to learn one classifier in each view; then, they bootstrap the views from each other by augmenting the training set with unlabeled examples on which the other views make high-confidence predictions. Such algorithms improve the classifiers learned from labeled data by also exploiting the implicit' information provided by the distribution of the unlabeled examples.

In contrast to semi-supervised learning, active learners (Tong & Koller, 2001) typically detect and ask the user to label only the most informative examples in the domain, thus reducing the user's data-labeling burden. Note that active and semi-supervised learners take different approaches to reducing the need for labeled data; the former explicitly search for a minimal set of labeled examples from which to perfectly learn the target concept, while the latter aim to improve a classifier learned from a (small) set of labeled examples by exploiting some additional unlabeled data.

In keeping with the active learning approach, this article focuses on minimizing the amount of labeled data without sacrificing the accuracy of the learned classifiers. We begin by analyzing co-testing (Muslea, 2002), which is a novel approach to active learning. Co-testing is a multi-view active learner that maximizes the benefits of labeled training data by providing a principled way to detect the most informative examples in a domain, thus allowing the user to label only these.

Then, we discuss two extensions of co-testing that cope with its main limitations—the inability to exploit the unlabeled examples that were not queried and the lack of a criterion for deciding whether a task is appropriate for multi-view learning. To address the former, we present Co-EMT (Muslea et al., 2002a), which interleaves co-testing with a semi-supervised, multi-view learner. This hybrid algorithm combines the benefits of active and semi-supervised learning by detecting the most informative examples, while also exploiting the remaining unlabeled examples. Second, we discuss Adaptive View Validation (Muslea et al., 2002b), which is a meta-learner that uses the experience acquired while solving past learning tasks to predict whether multi-view learning is appropriate for a new, unseen task.

A Motivating Problem: Wrapper Induction

Information agents such as Ariadne (Knoblock et al., 2001) integrate data from pre-specified sets of Web sites so that they can be accessed and combined via database-like queries. For example, consider the agent in Figure 1, which answers queries such as the following:

Show me the locations of all *Thai* ***restaurants in*** *L.A.* ***that are*** *A-rated* ***by the L.A. County Health Department.***

To answer this query, the agent must combine data from several Web sources:

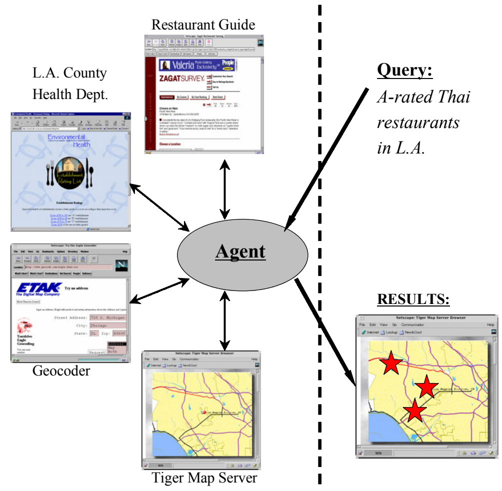
- from Zagat's, it obtains the name and address of all Thai restaurants in L.A.;
- from the L.A. County Web site, it gets the health rating of any restaurant of interest;
- from the Geocoder, it obtains the latitude/longitude of any physical address;
- from Tiger Map, it obtains the plot of any location, given its latitude and longitude.

Information agents typically rely on *wrappers* to extract the useful information from the relevant Web pages. Each wrapper consists of a set of extraction rules and the code required to apply them. As manually writing the extraction rules is a time-consuming task that requires a high level of expertise, researchers designed wrapper induction algorithms that learn the rules from user-provided examples (Muslea et al., 2001).

In practice, information agents use hundreds of extraction rules that have to be updated whenever the format of the Web sites changes. As manually labeling examples for each rule is a tedious, error-prone task, one must learn high accuracy rules from just a few labeled examples. Note that both the small training sets and the high accuracy rules are crucial to the successful deployment of an agent. The former minimizes the amount of work required to create the agent, thus making the task manageable. The latter is required in order to ensure the quality of the agent's answer to each query: when the data from multiple sources is integrated, the errors of the corresponding extraction rules get compounded, thus affecting the quality of the final result; for instance, if only 90% of the Thai restaurants and 90% of their health ratings are extracted correctly, the result contains only 81% (90% x 90% = 81%) of the A-rated Thai restaurants.

We use wrapper induction as the motivating problem for this article because, despite the practical importance of learning accurate wrappers from just a few labeled examples, there has been little work on active learning for this task. Furthermore, as explained in Muslea (2002), existing general-purpose active learners cannot be applied in a straightforward manner to wrapper induction.

Figure 1. An information agent that combines data from the Zagat's restaurant guide, the L.A. County Health Department, the ETAK Geocoder, and the Tiger Map service



MAIN THRUST

In the context of wrapper induction, we intuitively describe three novel algorithms: Co-Testing, Co-EMT, and Adaptive View Validation. Note that these algorithms are *not* specific to wrapper induction, and they have been applied to a variety of domains, such as text classification, advertisement removal, and discourse tree parsing (Muslea, 2002).

Co-Testing: Multi-View Active Learning

Co-Testing (Muslea, 2002, Muslea et al., 2000), which is the first multi-view approach to active learning, works as follows:

- first, it uses a small set of labeled examples to learn one classifier in each view;
- then, it applies the learned classifiers to all unlabeled examples and asks the user to label one of the examples on which the views predict different labels;
- it adds the newly labeled example to the training set and repeats the whole process.

Intuitively, Co-Testing relies on the following observation: if the classifiers learned in each view predict a different label for an unlabeled example, at least one of them makes a mistake on that prediction. By asking the user to label such an example, Co-Testing is guaranteed to provide useful information for the view that made the mistake.

Related Content