

Active Learning with Multiple Views

Ion Muslea

SRI International

AI Center, Office EJ-298

333 Ravenswood

Menlo Park, CA 94025

USA

Phone: (650) 859-5890

Fax: (650) 859-3735

Email: ion.muslea@sri.com

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 may also 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 chapter 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 sub-sets 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 chapter 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 and Singer, 1999, Pierce and 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 and 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 chapter 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.*, 2002.a), 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.*, 2002.b), 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 it can be accessed and combined via database-like queries. For example, consider the agent in Figure 1, which answers queries such as

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 ETAK 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 later 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 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% ($90\% \times 90\% = 81\%$) of the **A-rated Thai restaurants**.

We use wrapper induction as the motivating problem for this chapter 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, pp. 48-49), existing general-purpose active learners cannot be applied in a straightforward manner to wrapper induction.

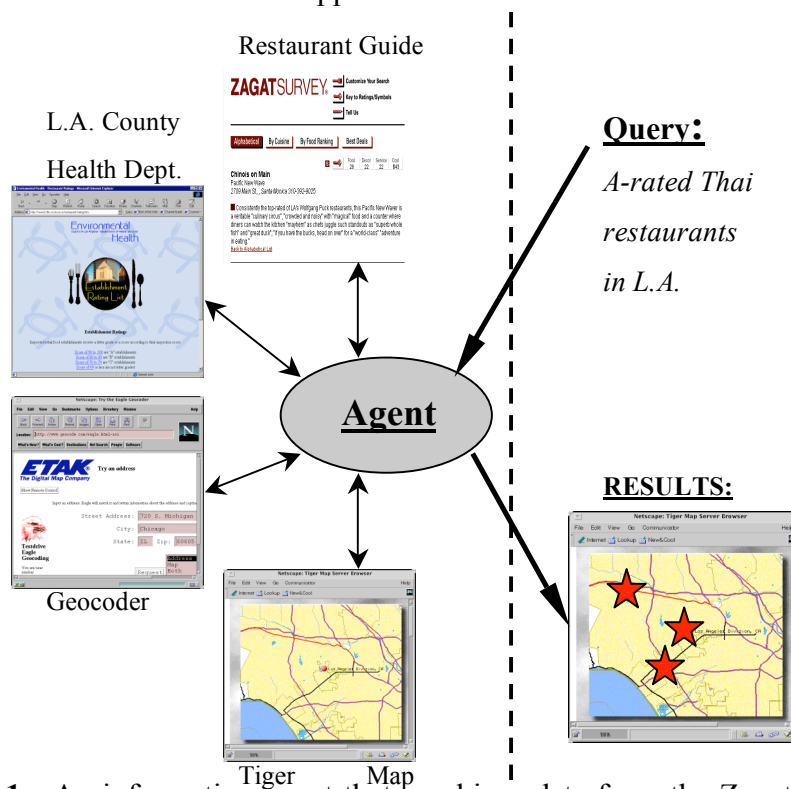


Figure 1: An information agent that combines data from the Zagat's restaurant guide, the L.A. County Health Department, the ETAK Geocoder, and the Tiger Map service.

MAIN THRUST OF THE CHAPTER

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.

To illustrate Co-Testing for wrapper induction, consider the task of extracting restaurant phone numbers from documents similar to one shown in Figure 2. To extract this information, the wrapper must detect both the beginning and the end of the phone number. For instance, to find where the phone number begins, one can use the rule

$$\mathbf{R1} = \textit{SkipTo}(\mathbf{Phone}:\langle i \rangle)$$

This rule is applied *forward*, from the beginning of the page, and it ignores everything until it finds the string **Phone**:<i>. Note that this is not the only way to detect where the phone number begins. An alternative way to perform this task is to use the rule

$$\mathbf{R2} = \textit{BackTo}(\mathbf{Cuisine}) \textit{BackTo}(\langle \mathbf{Number} \rangle)$$

which is applied *backward*, from the end of the document. **R2** ignores everything until it finds "cuisine" and then, again, skips to the first number between parentheses.

Note that **R1** and **R2** represent descriptions of the same concept (i.e., beginning of

phone number) that are learned in two different views (see (Muslea *et al.*, 2001) for details on learning forward and backward rules). That is, the views **V1** and **V2** consist of the sequences of characters that *precede* and *follow* the beginning of the item, respectively. The view **V1** is called the *forward view*, while **V2** is the *backward view*. Based on **V1** and **V2**, Co-Testing can be applied in a straightforward manner to wrapper induction. As shown in (Muslea, 2002), Co-Testing clearly outperforms existing state-of-the-art algorithms, both on wrapper induction and a variety of other real world domains.



Figure 2: The forward rule R1 and the backward rule R2 detect the beginning of the phone number. Forward and backward rules have the same semantics and differ only in terms of where they are applied from (start/end of the document) and in which direction.

Co-EMT: interleaving active and semi-supervised learning

To further reduce the need for labeled data, Co-EMT (Muslea *et al.*, 2002.a) combines active and semi-supervised learning by interleaving Co-Testing with Co-EM (Nigam and Ghani, 2000). Co-EM, which is a semi-supervised, multi-view learner, can be seen as an iterative, 2-step process: first, it uses the hypotheses learned in each view to probabilistically label all the unlabeled examples; then it learns a new hypothesis in each view by training on the probabilistically labeled examples provided by the other view.

By interleaving active and semi-supervised learning, Co-EMT creates a powerful synergy. On one hand, Co-Testing boosts Co-EM's performance by providing it with highly informative labeled examples (instead of random ones). On the other hand, Co-EM provides Co-Testing with more accurate classifiers (learned from both labeled and unlabeled data), thus allowing Co-Testing to make more informative queries.

Co-EMT was not yet applied to wrapper induction because the existing algorithms are

not probabilistic learners; however, an algorithm similar to Co-EMT was applied to information extraction from free text (Jones *et al.*, 2003). To illustrate how Co-EMT works, we describe now the generic algorithm Co-EMT^{WI}, which combines Co-Testing with the semi-supervised wrapper induction algorithm described below.

In order to perform semi-supervised wrapper induction, one can exploit a third view, which is used to evaluate the confidence of each extraction. This new, *content-based view* (Muslea *et al.*, 2003) describes the actual item to be extracted. For example, in the phone numbers extraction task, one can use the labeled examples to learn a simple grammar that describes the field content: "(*Number*) *Number* - *Number*". Similarly, when extracting URLs, one can learn that a typical URL starts with the string "`http://www.`", ends with the string "`.html`", and contains no HTML tags.

Based on the forward, backward, and content-based views, one can implement the following semi-supervised wrapper induction algorithm. First, the small set of labeled examples is used to learn a hypothesis in each view. Then the forward and backward views feed each other with unlabeled examples on which they make high-confidence

extractions (i.e., strings that are extracted by either the forward or the backward rule and are also compliant with the grammar learned in the third, content-based view).

Given Co-Testing and the semi-supervised learner above, Co-EMT^{WI} combines them as follows. First, the sets of labeled and unlabeled examples are used for semi-supervised learning. Second, the extraction rules that are learned in the previous step are used for Co-Testing. After making a query, the newly labeled example is added to the training set and the whole process is repeated for a number of iterations. The empirical study in (Muslea *et al.*, 2002.a) shows that, for a large variety of text classification tasks, Co-EMT outperforms both Co-Testing and the three state-of-the-art semi-supervised learners considered in that comparison.

View Validation: are the views adequate for multi-view learning?

The problem of *view validation* is defined as follows: given a new, unseen multi-view learning task, how does a user choose between solving it with a multi- or a single- view algorithm? In other words, how does one know whether multi-view learning will outperform pooling all features together and applying a single-view learner? Note that

this question must be answered while having access to just a few labeled and many unlabeled examples: applying both the single- and multi-view active learners and comparing their relative performances is a self-defeating strategy because it doubles the amount of required labeled data (one must label the queries made by both algorithms).

The need for view validation is motivated by the following observation: while applying Co-Testing to dozens of extraction tasks, Muslea *et al.* (2002.b) noticed that the forward and backward views are appropriate for most, *but not all* of these learning tasks. This view adequacy issue is tightly related to the best extraction accuracy reachable in each view. Consider, for example, an extraction task in which the forward and backward rules lead to a high- and a low- accuracy rule, respectively. Note that Co-Testing is not appropriate for solving such tasks: by definition, multi-view learning applies only to tasks in which each view is *sufficient* for learning the target concept (obviously, the low-accuracy view is *insufficient* for accurate extraction).

To cope with this problem, one can use *Adaptive View Validation* (Muslea *et al.*, 2002.b), which is a meta-learner that uses the experience acquired while solving past

learning tasks to predict whether the views of a new, unseen task are adequate for multi-view learning. The view validation algorithm takes as input several solved extraction tasks that are *labeled* by the user as having views that are *adequate* or *inadequate* for multi-view learning. Then it uses these solved extraction tasks to learn a classifier that, for new, unseen tasks, predicts whether the views are adequate for multi-view learning.

The (meta-) features used for view validation are properties of the hypotheses that, for each solved task, are learned in each view: the percentage of unlabeled examples on which the rules extract the same string, the difference in the complexity of the forward and backward rules, the difference in the errors made on the training set, etc. For both wrapper induction and text classification, Adaptive View Validation makes accurate predictions based on a modest amount of training data (Muslea *et al.*, 2002.b).

FUTURE TRENDS

There are several major areas of future work in the field of multi-view learning. First, there is a need for a *view detection* algorithm that *automatically* partitions a domain's features in views that are adequate for multi-view learning. Such an algorithm would

remove the last stumbling block against the wide applicability of multi-view learning: the requirement that the user provides the views to be used. Second, in order to reduce the computational costs of active learning (re-training after each query is CPU intensive), one must consider "look-ahead" strategies that detect and propose (near) optimal sets of queries. Finally, Adaptive View Validation has the limitation that it must be trained *separately* for each application domain (e.g., once for wrapper induction, once for text classification, etc). A major improvement would be a *domain independent* view validation algorithm that, once trained on a mixture of tasks from various domains, can be applied to any new learning task, independently of its application domain.

CONCLUSION

In this chapter, we focused on three recent developments that - in the context of multi-view learning - reduce the need for labeled training data:

- Co-Testing: a general-purpose, multi-view active learner that outperforms existing approaches on a variety of real world domains.

- Co-EMT: a multi-view learner that obtains a robust behavior over a wide spectrum of learning tasks by interleaving active and semi-supervised multi-view learning.
- Adaptive View Validation: a meta-learner that uses past experiences to predict whether multi-view learning is appropriate for a new, unseen learning task.

REFERENCES

- Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Conference on Computational Learning Theory (COLT-1998)*, 92-100.
- Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. *Empirical Methods in Natural Language Processing & Very Large Corpora*, 100-110.
- Jones, R., Ghani, R., Mitchell, T., & Riloff, E. (2003). Active learning for information extraction with multiple view feature sets. The ECML-2003 Workshop on *Adaptive Text Extraction and Mining*, <http://www.dcs.shef.ac.uk/~fabio/ATEM03/accepted.html>
- Knoblock, C., Minton, S., Ambite, J.-L., Ashish, N., Muslea, I., & Philpot, A. (2001).

- The Ariadne approach to Web-based Information Integration. *International Journal of Cooperative Information Sources*, 10, 145-169.
- Muslea, I. (2002). Active Learning with Multiple Views. Ph.D. thesis, Department of Computer Science, University of Southern California.
- Muslea, I., Minton, S., & Knoblock, C. (2000). Selective sampling with redundant views. *National Conference on Artificial Intelligence (AAAI-2000)*, 621-626.
- Muslea, I., Minton, S., Knoblock, C. (2001). Hierarchical wrapper induction for semi-structured sources. *Journal of Autonomous Agents & Multi-Agent Systems*, 4, 93-114
- Muslea, I., Minton, S., & Knoblock, C. (2002a). Active + Semi-supervised Learning = Robust Multi-view Learning. *International Conference on Machine Learning (ICML-2002)*, 435-442.
- Muslea, I., Minton, S., & Knoblock, C. (2002b). Adaptive view validation: A first step towards automatic view detection. *International Conference on Machine Learning (ICML-2002)*, 443-450.

- Muslea, I., Minton, S., & Knoblock, C. (2003). Active learning with strong and weak views: a case study on wrapper induction. *International Joint Conference on Artificial Intelligence (IJCAI-2003)*, 415-420.
- Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Information and Knowledge Management (CIKM-2000)*, 86-93.
- Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3), 103-134.
- Pierce, D., & Cardie, C. (2001). Limitations of co-training for natural language learning from large datasets. *Empirical Methods in Natural Language Processing*, 1-10.
- Raskutti, B., Ferra, H., & Kowalczyk, A. (2002). Using unlabeled data for text classification through addition of cluster parameters. *International Conference on Machine Learning (ICML-2002)*, 514-521.
- Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2, 45-66.

TERMS AND THEIR DEFINITION

Inductive learning: acquiring concept descriptions from labeled examples.

Multi-view learning: explicitly exploiting several disjoint sets of features, each of which is sufficient to learn the target concept.

Active learning: detecting and asking the user to label only the most informative examples in the domain (rather than randomly chosen examples).

Semi-supervised learning: learning from both labeled and unlabeled data.

Meta-learning: learning to predict the most appropriate algorithm for a particular task.

View Validation: deciding whether a set of views is appropriate for multi-view learning.

Wrapper induction: learning (highly accurate) rules that extract data from a collection of documents that share a similar underlying structure.