Empowered Learners!

Omid Madani
omid.madani@overture.com

June 11, 2004

In the standard supervised learning setting the learner is given a set of labeled examples and asked to learn a function/model/classifier that can correctly label future unlabeled *test* examples. The objective is to minimize a measure of misclassification error on the future examples (not seen at training time). The set of training labeled instances/examples is provided in a single batch, and once learned, the test examples are seen one after another, and the test examples are drawn independently of one another and follow the same distribution as the distribution used to obtain the labeled training set.

An example of this is webpage classification of university pages into say department page, professor pages, and student pages, etc. At training time, a set of pages all labeled is given, and the classifier induced takes as input a single page and outputs the label.

All the following learning frameworks assume additional inputs, resources, or capabilities for the learner, and offer the potential for reducing costs in labeling data and/or superior learner accuracy.

1 Active Learning

In typical "pool-based" active learning, the learner also has access to a set of unlabeled instances, and it can ask a teacher for the label of a small subset of those examples. The rough objective is to ask as few questions as possible and ramp up the classifier accuracy as fast as possible. One precise objective would be the learner is given a budget M, so it can ask for the labels of at most M unlabeled instances. It is assumed the learner can ask its questions sequentially (ask after finding out about the label of the previous query). So the problem is to find an algorithm that can do the best achievable accuracy subject to the budget. For an interesting somewhat different/experimental way of evaluating active learning algorithms, see [BEYL03].

In the webpage example, an active learning setting would involve the learner being given a corpus of many unlabeled pages, and it would then ask for the labels of a small subset of the pages selectively and dynamically.

Other forms of active learning include allowing the learner to construct its own (synthetic) example. Active learning is also referred to "query learning" in computational learning theory [Ang92, Gol99] (the focus has mostly been on exact learning of a concept from a concept class), and also selective sampling (perhaps in vision or stats literature?). I believe pool-based active learning has had, or appears to have, much more applications than other models of active learning. I think a first paper on this model (at least in the learning literature) is [CAL94] (who call it "query filtering" paradigm: the learner can sample from the instance distribution and decides whether or not to query the oracle/teacher for the label whenever it samples an instance).

Researchers have shown that active learning can significantly reduce the trainig-set size rquirements both in theory (*e.g.*, make an unlearnable/intractable learning problem under the standard model learnable), and in empirical settings. For example, Baum shows that using membership queries (with synthetic instances) makes possible efficient learning of neural nets with 4 hidden units [Bau91], otherwise intractable in theory. However, later work showed that synthesized instances may not be recognizable/labelable to humans in the task of handwritten character recognition[1] [BL92]. Perhaps for this reason, pool-based active learning has been researched more. In pool-based/query-filtering setting, Freund et al establish the potential for exponential reduction in the number of labels asked (*e.g.*, with each label query, the error is halved) as long as certain assumptions are made [FSST97]. The assumptions are roughly that "information gain" is lower bounded by a constant while the desired error level has not been reached, where the information gain is the reduction in entropy over the hypothesis class, or roughly reducing the number of possible models by a half each time a query is made (which implies reduction error by a constant fraction in their model). They show this property holds for linear linear classifiers. In mnay empirical studies, the savings are anywhere from requiring half to smaller fractions depending on the problem domain and desired error level.

Most (pool-based) active learning algorithms are "greedy" in the sense that they have a (one-step) criterion (eg committee disagreement [SOS92, FSST97], or a measure of uncertainty of the classifier, or *uncertainty sampling* [LG94]) and they pick the instance that optimizes it. The intuition behind uncertainty sampling and committee disagreement measure is that criterion quickly shrinks down the space of models that are consistent with the training set. However, they these criteria may suffer if absolute size of the outliters in the unlabeled set is large (while the proportion may still be low). In some cases, analytic closed-forms exist for picking the instance that's expected to improve the accuracy of the current classifier the most [CGJ96], and others have explored approximating this criterion [LMR02, RM01]. Could the learner do better if it knew it could ask 100 questions rather than acting as if it has

[1] This problem may be ameliorated if one also tried to learn a generative model of the data, but the if one has enough unlabeled data, synthesizing may not add extra power..

only one more question? See [LMR02] for a discussion, and they observed looking 2 plies deep didn't improve the active learning significantly (lowered variance over the accuracy however). In a different active learning problem (in model selection, and also where feature values are unknown), we observe "looking-deeper" (in the search tree) is necessary for superior performance [MLG04, LMG03].

Other related problems (besides better active learners in various contexts) are error estimation and/or determining when to stop (say instead of a budget, we want to stop when we know we can do no better, or it's not worth the bang for the buck, or we have reached satisfactory accuracy, etc.) (an application of our recent work touches on this [MPF]). In general, I think we need to better understand the power of active learning and its limits in statistical settings (*e.g.*, [CC95, ZO00]), and there remains a number of empirical and theoretical questions. For example, how is the effectiveness of active learning related to the number of irrelevant features (or ratio of relevant to irrelevant). For instance, what if an oracle told you the set of relevant (necessary and sufficient) features? I am also interested in various ways of giving hints/feedback to the (active) learner (*e.g.*, not just labels for instances, but also choosing samples likely to contain positives, and indications of relevancy of features, see eg [Jon04] for labeling features).

2 Semi-Supervised Learning

I believe semi-supervised learning is mostly used to refer to the following scenario: the learner has labeled and unlabeled data available, and the question is whether the learner can do better, *e.g.*, can generalize better to *unseen* data, by using the unlabeled data as well as the labeled ones. Typically, one assumes the learner is not allowed to ask for the labels of the unlabeled instances, though there is at least one work that explores semi-supervised techniques with limited active learning. One idea for using unlabeled data is to use them for some form of bootstrapping: *e.g.*, label (a subset of) the unlabeled instances with the current classifier(s), retrain with the larger labeled set, and repeat. Another major use of unlabeled data is in estimation of other aspects of the classifier (eg. its error) and model selection (any other different uses?). Major semi-supervised techniques/ideas include:

- coTraining: roughly, if you have two feature sets (or two or more classifiers) that work independently, then each classifier trained on one set can be used to train the other by labeling those unlabeled instances that it's most confident on [BM98].
- Other bootstrapping methods, such as EM, *e.g.*, [NMTM00].
- Clustering methods, for example clustering the score of the classifier on the unlabeled data to choose a better threshold for the classifier.

- Using unlabeled data for model selection, regularization/controlling the complexity of the model, and error estimation in active learning, etc.*e.g.*, [KV95, SS02, MPF].

See [See01] for a survey on using unlabeled data in semi-suprvised learning.

3 Transduction

The term transduction is credited to Vapnik: in transduction, one has labeled data and unlabeled data, and the objective is to label the *seen* unlabeled as best as possible. So the difference with induction and semi-supervised learning is that the learner gets to see the instances it's tested on (like a take-home exam, or cheating..!). So whenever you get to see the test instances and you optimize/estimate various parameters using the test data (for example, choosing the threshold for a linear classifier), that's really transduction! In the web-page classification example, the learner would be given a database of the pages (say all the university pages in the world), in which a relatively small subset is labeled, and is asked to label the remainder as accurately as possible. Applications include labeling items in a (relatively static) database. (also it may be possible to use transduction for bootstrapping in semi-supervised learning).

Successful transduction techniques have so far involved some sort of clustering or graph partitioning: *e.g.*, clustering the instances, and labeling a clusters' unlabeled members according to majority label of the labeled instances in that cluster [Joa03, ZBL+03, DEYM03]. This assumes we have a reliable similarity metric for clustering and that the clustering algorithm is appropriate for the learning task as well.

Currently, it appears that the biggest bang for the buck (for the domains tested on in literature) comes when we have a relatively small set of labeled instances. As the number of labeled instances grows to more than 10s say, the benefit of transduction (using current techniques) over pure induction (*i.e.*, just using labeled data to train the classifier ignoring the unlabeled, and then use the classifier to label the unlabeled) become negligible. It would be great to quantify when/how much can transduction help (as a function of complexity of the learning problem, etc., *e.g.*, see [ZO00]). My current feeling is that the current techniques are not robust enough and the range of their applicability is somewhat limited (so room for more research). However, the work of El-Yaniv et. al. [DEYM03] is particularly interesting as they derive pretty tight bounds on the transductive error. Such bounds are not currently available in the induction setting. This ability to estimate error better may be another lasting benefit of transduction over induction.

4 Relational Learning

In one version of relational learning, the learner classifies the instances in "bunches", and the instances have relations to one-another in a way that constrains their mutual labelings (eg characters in streams of words in character recognition, or image segmentation, or the classification of various objects in an image, classification of various parts before recognizing the whole, etc.). So the learner could in theory use this extra information, *reason* with it, and come up with a better overall labeling (*e.g.*, [TWAK03, TGK03]). Note that the iid assumptions of typical inductive learning (instances are independent and identically distributed) no longer hold. In another version of relational learning, the features can be relational, for example [PULP03], and the challenges include sifting through thousands of relational features to find the best few features.

In the web-page classification example, the problem may be given in the form of a directed graph where the labels of vertices affect one another (*e.g.*, faculty pages are likely to link to research and students pages).

One challenge is that typically relational learning methods are significantly more demanding computationally than their flat counterparts due to their reasoning components. Roth et al[KR97] argue that one should base reasoning (which has often been intractable except for simplest setting) on learning (moreover on simple flat propositional and even just linear learning components for the most part). In any case, the general area of studying the interaction of learning and inference (of which relational learning is an incarnation) is still a wide open area in many respects. It should have great impact in the long run.

5 Discussion

Various combinations are possible: eg. active learning for transduction or for relational learning. The transduction technique of clustering maybe viewed a technique of relational learning: the label of an instance is not solely determined by the labeled instances, but its relation (*e.g.*, proximity) to unlabeled instances and their likely labels as well.

All the above learning settings are promising. They are fairly new and often poorly understood, and not as robust as their mother: classic inductive learning. They remain an exciting area of research.

References

[Ang92] D. Angluin. Computational learning theory: survey and selected bibliography. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages

351–369. ACM Press, New York, NY, 1992.

[Bau91] E. Baum. Neural network algorithms that learn in polynomial time from examples and queries. *IEEE Trans. Neural Networks*, 1991.

[BEYL03] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. In *ICML*, pages 19–26, 2003.

[BL92] E. Baum and K. Lang. Query learning can work poorly when a human oracle is used. In *International Joint Conference in Neural Networks*, 1992.

[BM98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[CAL94] D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[CC95] V. Castelli and T. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 1995.

[CGJ96] D. Cohn, Z. Gharamani, and M. I. Jordan. Active learning with statistical model. *Journal of Artificial Research*, 1996.

[DEYM03] P. Derbeko, R. El-Yaniv, and R. Meir. Error bounds for transductive learning via compression and clustering. In *NIPS*, 2003.

[FSST97] Y. Freund, H. Sebastian Seung, E. Shamir, and N. Tishby. Selective sampling usign the query by committee algortihm. *Machine Learning*, 1997.

[Gol99] S. Goldman. Computational learning theory. In *Algorithms and Theory of Computation Handbook*. CRC Press, 1999.

[Joa03] T. Joachims. Transductive learning via spectral graph partitioning. In *290-297*, 2003.

[Jon04] R. Jones. *Bootstrapping Algorithms and Active Learning for Minimally-supervised Machine Learning of Semantic Classes with Redundant Feature Sets*. PhD thesis, CMU, 2004. in preparation.

[KR97] R. Khardon and D. Roth. Learning to reason. *JACM*, 1997.

[KV95] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *NIPS*, 1995.

[LG94] D. Lewis and W. Gale. Training text classifiers by uncertainty sampling. In *SIGIR*, 1994.

[LMG03] D. J. Lizotte, O. Madani, and R. Greiner. Budgeted learning of Naive Bayes classifiers. In *Uncertainty in AI*, 2003. http://www.cs.ualberta.ca/∼madani/budget.html.

[LMR02] M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 2002.

[MLG04] O. Madani, D. Lizotte, and R. Greiner. Active model selection (to appear in uai04). Technical report, University of Alberta and AICML, 2004. http://www.cs.ualberta.ca/∼madani/budget.html.

[MPF] O. Madani, D. M. Pennock, and G. W. Flake. Co-validation: Using model disagreement to validate classification algorithms. Manuscript. Presented at Snowbird, 2004.

[NMTM00] K. Nigham, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 2000.

[PULP03] A. Popescul, L. H. Ungar, S. Lawrence, and D .M. Pennock. Towards structural logistic regression: Combining relational and statistical learning. In *In Proceedings of IEEE International Conference on Data Mining (ICDM)*, 2003.

[RM01] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001.

[See01] M. Seeger. Learning with labeled and unlabeled data. Technical report, Edingbugh University, 2001.

[SOS92] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, pages 287–294, 1992.

[SS02] D. Schuurmans and F. Southey. Metric based methods for adaptive model selection and regularization. *Machine Learning*, 48(1-3):51–84, 2002.

[TGK03] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, 03.

[TWAK03] B. Taskar, M. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *NIPS*, 03.

[ZBL+03] D. Zhou, O. Bousquet, T N. Lal, J. Weston, and B. Schoelkopf. Learning with local and global consistency. In *NIPS*, 2003.

[ZO00] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML 2000*, pages 1191–1198, 2000.