Active Learning for Pipeline Models

Dan Roth and Kevin Small
University of Illinois at Urbana-Champaign
201 N Goodwin Ave
Urbana, IL 61801 USA
{danr,ksmall}@uiuc.edu

Decomposing complex classification tasks into a series of sequential stages, where the local classifier at each stage is explicitly dependent on predictions from previous stages, is a common practice. In the machine learning and natural language processing communities, this widely used paradigm is generally referred to as a pipeline model. This approach has been successfully applied to many tasks, including parsing, semantic role labeling, and textual entailment [3]. The primary motivation for modeling complex tasks as a pipelined process is the difficulty of solving such applications with a single classifier; that learning a classifier for a problem such as relation extraction directly in terms of input text may be impossible with the given resources.

A second feature of domains requiring such decompositions is the corresponding high cost associated with obtaining sufficient labeled data. The active learning protocol offers one promising solution to this dilemma by allowing the learning algorithm to incrementally select unlabeled examples for labeling by the domain expert with the goal of maximizing performance while minimizing supervision [1]. While receiving significant recent attention, most active learning research focuses on new algorithms as they relate to a single classification task. This work instead assumes that an active learning algorithm exists for each stage of a pipelined model and develops a strategy that jointly minimizes the annotation requirements for the pipelined process.

Consider a D-stage pipeline where $x \in \mathcal{X}$ represents input instances, $\Phi^{(d)}(x, \hat{y}^{(0)}, \ldots, \hat{y}^{(d-1)}) \rightarrow \mathbf{x}^{(d)}$ represents the feature vector generating function of the d^{th} stage, and $y^{(d)} \in \mathcal{Y}^{(d)}$ represents the corresponding label. Each stage of a pipelined learning process takes m training instances $\mathcal{S}^{(d)} = \{(\mathbf{x}_1^{(d)}, y_1^{(d)}), \ldots, (\mathbf{x}_m^{(d)}, y_m^{(d)})\}$ as input to a learning algorithm $\mathcal{A}^{(d)}$ and returns a classifier, $h^{(d)}$, based upon a scoring function, $f^{(d)}$, which minimizes the respective loss function of the d^{th} stage. Each stage may vary in complexity from a binary prediction, $y^{(d)} \in \{-1, 1\}$, to a multiclass prediction, $y^{(d)} \in \{\omega_1, \ldots, \omega_k\}$, to a structured output prediction, $y \in \mathcal{Y}_1^{(d)} \times \cdots \times \mathcal{Y}_{n_y}^{(d)}$. Once each stage of the pipeline model classifier is learned, global predictions are made sequentially with the goal of maximizing performance on the overall task,

$$\hat{\mathbf{y}} = h(x) = \left\{ \underset{y' \in \mathcal{Y}^{(d)}}{\operatorname{argmax}} f^{(d)} \left(\mathbf{x}^{(d)}, y' \right) \right\}_{d=1}^{D} . \tag{1}$$

The key difference between the active learning and standard supervised learning protocols is a querying function, $\mathcal{Q}$, which uses the data $\mathcal{S}$ and the current learned classifier h to return unlabeled instances $\mathcal{S}_{select} \subseteq \mathcal{S}_u$ which are labeled and added to $\mathcal{S}_l$. This paper assumes an underlying query scoring function, $q : x \rightarrow [0, 1]$, for each stage of a pipeline model such that the instance with the minimum query scoring function is selected for labeling, $\mathcal{Q} : x_\star = \text{argmin}_{x \in \mathcal{S}_u} q(x)$. While notation requires only x to return a score, we assume that $q(x)$ implicitly has access to required facilities to make this determination (e.g. f, h, Φ, etc.). Given a query scoring function for each stage of the pipeline, we generate a joint ranking function of the functional form

$$\mathcal{Q}_{pipeline} : x_\star = \underset{x \in \mathcal{S}_u}{\text{argmin}} \sum_{d=1}^{D} \beta^{(d)} \cdot q^{(d)}(x) \tag{2}$$

where $\beta^{(d)}$ is adapted for each active learning iteration to adjust the relative impact of the query function for each stage. Extending the work in [2] for single predictions, we derive a procedure which determines $\beta^{(d)}$ for each iteration based on the relative change of average uncertainty over the remaining unlabeled examples for each stage.

We demonstrate our active learning approach on a three-stage entity and relation extraction task comprised of a segmentation stage, an entity classification stage, and a relation classification stage. We design active learning algorithms for each stage of the pipeline based on the principles outlined in [4] for structured output problems and mix the relative impact of the three query scoring functions according to our joint active learning protocol. Using F1 as our performance measure, we show a 47% reduction in annotation requirements for segmentation, 46% reduction in annotation requirements for entity classification, and a 35% reduction in the annotation effort for relation classification. This method significantly outperforms both a random selection criteria and an active learning criteria for structured output spaces that ignores the pipeline structure.

References

[1] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[2] Pinar Donmez, Jaime G. Caronell, and Paul N. Bennett. Dual strategy active learning. In *Proc. of the European Conference on Machine Learning (ECML)*, 2007.

[3] Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

[4] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Proc. of the European Conference on Machine Learning (ECML)*, 2006.

Topic: Active Learning, Pipeline Models
Preference: Oral