Active Learning amidst Logical Knowledge

Emmanouil A. Platanios [1] **Ashish Kapoor** [2] **Eric Horvitz** [2]

Abstract

Structured prediction is ubiquitous in applications of machine learning such as knowledge extraction and natural language processing. Structure often can be formulated in terms of logical constraints. We consider the question of how to perform efficient active learning in the presence of logical constraints among variables inferred by different classifiers. We propose several methods and provide theoretical results that demonstrate the inappropriateness of employing uncertainty guided sampling, a commonly used active learning method. Furthermore, experiments on ten different datasets demonstrate that the methods significantly outperform alternatives in practice. The results are of practical significance in situations where labeled data is scarce.

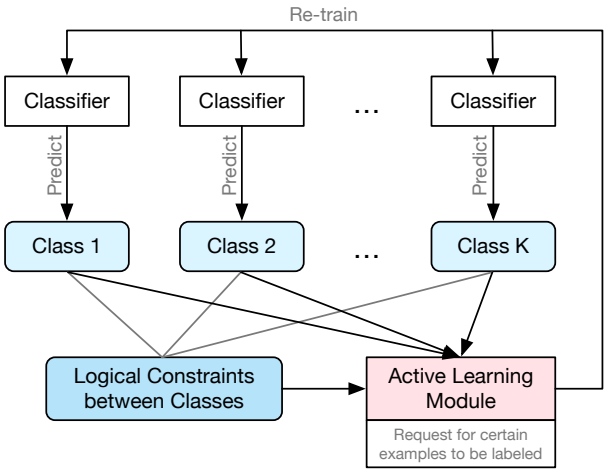

Figure 1: Illustration of active learning in an interdependent multiple classifier setting.

1. Introduction

Tasks which involve learning several classifiers whose outputs are tied together by logical constraints are abundant in machine learning. As an example, we may have two classifiers in the Never Ending Language Learning (NELL) project (Mitchell et al., 2015) which predict whether noun phrases represent animals or cities, respectively. In this case, the outputs of the two classifiers are mutually exclusive. Many such tasks hinge on the training of a large number of classifiers in situations where obtaining labeled data is expensive. The difficulty of acquiring labels leads to the common approach (highlighted in Figure 1) of performing an initial training of classifiers with a small number of labeled examples, and then iteratively identifying the most valuable additional labels to acquire, followed by the re-training of the classifiers. We seek methods that are capable of performing such *active learning* (Settles, 2012), an instance of *semi-supervised learning*. In this paper, we propose methods for active learning that share a common underlying goal: efficient identification of the most valuable labels to acquire in the presence of *logical constraints* among the outputs of classifiers being trained. Examples of such constraints are mutual exclusion (e.g., in multi-class/one-vs-all classification) and subsumption (e.g., in hierarchical classification) among target variables. In active learning for mutual exclusion and subsumption, we need to consider the complexities of behavior arising in the interactions among the linked classifiers. We shall provide theoretical justification for the proposed methods that resonates with intuition. As we will show, the results challenge the core idea behind *uncertainty guided sampling*, a method in common practice.

We motivate our work with challenges in information extraction, where noun phrases are mapped to various categories (e.g., `animal`, and `bird`) and relations (e.g., `animalEatsFood`). It is easy to see how these categories and relations can be tied through logical constraints. For example, one might say that `animal` and `location` are mutually exclusive, and `animal` subsumes `bird`. We consider examples highlighted by work on the NELL project (Mitchell et al., 2015). NELL currently performs over 2,500 learning tasks and it is thus too expensive to obtain enough labeled data for each task separately. The ability to rank examples by the utility of discovering their labels would enable the system to more efficiently allocate and use resources available for labeling. Our goal in this paper is to provide systems such as NELL with this ability, such that their learning rate would be significantly increased with respect to the resources available for labeling.

[1]Carnegie Mellon University, Pittsburgh, PA, USA (research performed during an internship at Microsoft Research) [2]Microsoft Research, Redmond, WA, USA. Correspondence to: Emmanouil A. Platanios <e.a.platanios@cs.cmu.edu>.

2. Related Work

The literature covers many projects in the realms of active learning (Settles, 2012; Ruvolo and Eaton, 2013) and decision theory (e.g., the core concept of *value of information* and its use in guiding the collection of examples) (Kapoor et al., 2007; Krause and Guestrin, 2009). Related work on computing the value of information for inference that leverages structural information includes an effort showing how the topology of influence diagrams could be used to assert an ordering over the value of information for variables (Poh and Horvitz, 1996). However, most existing approaches to active information gathering for machine learning are directed at collecting single labels for one classifier. Furthermore, even approaches that deal with settings involving multiple labels do not make use of logical constraints that may exist among labels (Reichart et al., 2008; Zhao et al., 2015). Work in the area of semi-supervised learning makes it clear that such constraints are present in many practical settings and that they can indeed prove useful if used appropriately (Chang et al., 2007; 2008; Mitchell et al., 2015).

There have been a few approaches that make use of such constraints. First, we note that query-by-committee (QBC) can be viewed as a special case of our framework, where the logical constraint is that committee members must agree. Settles and Craven (2008) propose approaches to perform active learning for sequence labeling tasks, including uncertainty sampling and QBC. Culotta and McCallum (2005) consider adding constraints to such tasks. To the best of our knowledge, they are the first to consider general constraints. In distinction to this prior work, we do not focus on the difficulty of each labeling task. We consider a wider range of tasks. Culotta and McCallum present only one instantiation of our more general formulation. Luo et al. (2013) use uncertainty sampling, where probabilities are computed using classifiers that account for constraints. Roth and Small (2008) score instances using the margin of learned classifiers and in the case of binary classification, margin-based active learning is equivalent to uncertainty sampling. Bilgic et al. (2010) consider dependencies among input instances and their labels. They cluster instances and look at disagreements of two kinds of classifiers over the clusters. Other methods that use "side" information in active learning include those of Kapoor and Baker (2009), Wallace et al. (2010), and Angeli et al. (2014).

Our method considers the important and common case where there are logical constraints over the output space, such as mutual exclusion and subsumption. The ubiquitous nature of such logic relationships creates a need for them to be addressed "head on". The previously mentioned related work only deals with other kinds of probabilistic constraints. Harpale (2012) and Zhang (2010) have considered this setting, but they fail to provide theoretical justifications nor provide deep experimental support. Furthermore both approaches can be seen as separate instantiations of our more general framework, for which we also provide a formal analysis along with an extensive experimental evaluation.

3. Proposed Methods

We now provide a description of methods for performing active learning. The methods select examples to be labeled before each re-training step (i.e., the red box in Figure 1). Let us consider a setting where we have a set of binary labels $Y_k^i \in \{0, 1\}$, for $k = 1, \ldots, K$ and $i = 1, \ldots, N$, for a provided set of instances $X^1, \ldots, X^N$. Y_k^i denotes whether instance X^i belongs to class k. One example would be where X^i represents a particular noun phrase (NP) and Y_k^i is a particular label for that NP, indicating whether it is a city or not. There exists a set of logical constraints among the K labels for each instance which determine whether an assignment of values to those labels is valid or not. Let the marginal probability of each label being positive be defined as $p_k^i \triangleq \mathbb{P}_{X^i \sim \mathcal{D}}(Y_k^i = 1)$, for $k = 1, \ldots, K$ and $i = 1, \ldots, N$, where $\mathcal{D}$ is the distribution of the instances $X^1, \ldots, X^N$. Given a set of observed labels (which could be empty) and these marginal probabilities, we want to determine which label[1] to request in the active learning process in order to gain the most information. Thus, we use a *scoring function* to score each unobserved label based on how much information is gained by observing it, and we then pick the label with the highest such score. We note that *information gain* can be defined in many ways depending on the task at hand and the evaluation metric that is being used. Our approach is initially motivated by the loose and possibly naïve definition of information gain as the expected number of labels one obtains after asking for a single label (i.e., due to the constraints among the labels).

We note that the common strategy of *uncertainty guided sampling* for allocating labeling effort uses the entropy of a label as its scoring function. That is:

$$\mathcal{S}_{\text{entropy}}(Y_k^i) \triangleq -p_k^i \log p_k^i - (1 - p_k^i) \log(1 - p_k^i). \quad (1)$$

Thus, this function can be thought of as scoring each label based only on its own uncertainty ignoring any dependencies among the labels. Our proposed methods make use of logical constraints among the labels, thus considering key dependencies. We therefore expect them to perform better in practice.

3.1. A Simple Constraint: Mutual Exclusion

Let us first consider a simple, yet powerful and common logical constraint among labels: *mutual exclusion*. We consider a setting where, for each value of i (i.e., instance), all labels (i.e., $Y_1^i, \ldots, Y_K^i$) are mutually exclusive with one another. This means that, for each instance, at most one label can be positive. It is easy to see that, if we discover that a label for a specific instance is positive, then all other

[1] Note that the word "label" here refers to a particular label-instance pair (i.e., we ask for a single label of a single instance at a time). This is the convention we use throughout this paper.

labels must be negative for that instance. However, if the answer is negative, then we cannot infer the value of any other label. Thus, intuitively we see that it might make sense to ask for the label with the highest marginal probability of being equal to 1 (i.e., the Y_k^i with the highest probability p_k^i). We now discuss this approach and provide theoretical justification for this intuition. We start by suggesting the following scoring function:

$$\mathcal{S}_{\text{probability}}(Y_k^i) \triangleq p_k^i. \tag{2}$$

For the following theoretical justification, we shall ignore the instance superscript (i.e., i) and consider the case where there is only a single instance X. We shall propose a theorem related to this scoring function, but first we will state a lemma that will be used in the forthcoming proof:

Lemma 1. *Let $x \in [0,1]$, and $c \in [0, 1-x]$. Then, the following function is monotonic with respect to x: $f(x) = (1-x-c)\log(1-x-c) - (1-x)\log(1-x)$.*

Proof. We have that $\frac{\partial f(x)}{\partial x} = \log(1-x) - \log(1-x-c)$ and since the logarithm is a monotonic function, we know that $\frac{\partial f(x)}{\partial x} \geq 0$. Thus, $f(x)$ is monotonic. □

Theorem 1. *Given a set of mutually exclusive labels, the scoring function in Equation 2 induces the same ranking of labels as the information-theoretic information gain.*

Proof. Due to the mutual exclusion constraint, we have $\mathbb{P}_{X\sim\mathcal{D}}(\{Y_k = 0 \text{ for } k = 1,\ldots,K\}) = 1 - \sum_{k=1}^{K} p_k$. For notational convenience, let us denote this quantity by p_0 and also omit the $X \sim \mathcal{D}$ subscript from the probability operator notation henceforth. Now, note that:

$$\begin{aligned}\mathbb{P}(\boldsymbol{y_{-k}}) &= \mathbb{P}(\boldsymbol{y_{-k}} \wedge Y_k = 1) + \mathbb{P}(\boldsymbol{y_{-k}} \wedge Y_k = 0),\\ &= \begin{cases} 0 & \text{, if } \boldsymbol{y_{-k}} \text{ has more than one 1s,}\\ p_l & \text{, if } y_l = 1 \text{ for } l \neq k,\\ p_k + p_0 & \text{, otherwise,}\end{cases}\end{aligned}$$

where $\boldsymbol{y_{-k}}$ refers to an assignment of values to all labels Y_l, where $l = 1,\ldots,K$, and $l \neq k$, and y_l refers to an assignment of Y_l. Let us also denote the information-theoretic information gain of variable Y_k by $\mathcal{I}(Y_k)$. We then have that if $p_k \geq p_l$, for some $k \neq l$, then:

$$\begin{aligned}&\mathcal{I}(Y_k) - \mathcal{I}(Y_l)\\ &= \mathcal{H}(\boldsymbol{Y_{-k}}) - \mathcal{H}(\boldsymbol{Y_{-k}} \mid Y_k) - \mathcal{H}(\boldsymbol{Y_{-l}}) + \mathcal{H}(\boldsymbol{Y_{-l}} \mid Y_l),\\ &= \mathcal{H}(\boldsymbol{Y_{-k}}) + \mathcal{H}(Y_k) - \mathcal{H}(\boldsymbol{Y_{-l}}) - \mathcal{H}(Y_l),\\ &= -\sum_{\boldsymbol{y_{-k}}} \mathbb{P}(\boldsymbol{y_{-k}})\log\mathbb{P}(\boldsymbol{y_{-k}}) - p_k\log p_k,\\ &\quad + \sum_{\boldsymbol{y_{-l}}} \mathbb{P}(\boldsymbol{y_{-l}})\log\mathbb{P}(\boldsymbol{y_{-l}}) + p_l\log p_l,\\ &\quad - (1-p_k)\log(1-p_k) + (1-p_l)\log(1-p_l),\\ &= (p_l+p_0)\log(p_l+p_0) - (1-p_k)\log(1-p_k),\\ &\quad - (p_k+p_0)\log(p_k+p_0) + (1-p_l)\log(1-p_l),\\ &= (1-p_k-c)\log(1-p_k-c) - (1-p_k)\log(1-p_k),\\ &\quad - (1-p_l-c)\log(1-p_l-c) + (1-p_l)\log(1-p_l),\\ &\geq 0,\end{aligned}$$

where $\mathcal{H}(Y_k)$ corresponds to the entropy of the Y_k variable, $\mathcal{H}(\boldsymbol{Y_{-k}})$ corresponds to the entropy of all variables Y_l, where $l = 1,\ldots,K$ and $l \neq k$, and $H(p) \triangleq -p\log p - (1-p)\log(1-p)$. The sums are over all possible assignments of the corresponding variables. The last step follows from Lemma 1, where $c = \sum_{k'=1,k'\neq k,l}^{K} p_{k'}$. The above inequality implies that the ranking of labels induced by the information gain $\mathcal{I}(Y_k)$ is the same as the ranking induced by using the scoring function $\mathcal{S}_{\text{probability}}(Y_k^i)$ and the proof is complete. □

One of the most interesting consequences of Theorem 1 is that we now have a very efficient way to rank labels based on their information gain. Also note that more often than not, classification systems are evaluated based on the area under the precision-recall curve (AUC). Furthermore, one might care about maximizing the number of "gold" labels, meaning labels that are guaranteed to be correct. Intuitively, the AUC increases with the number of "gold" labels. We highlight the fact that the probability scoring function of Equation 2 was motivated by picking the label that is most likely to provide the greatest number of "gold" labels (i.e., labels that are fixed to 0).

We note that the scoring function $\mathcal{S}_{\text{probability}}(Y_k^i)$ assigns higher score to labels that are more certainly positive than to more uncertain labels. This is in contrast to uncertainty guided sampling and thus highlights the importance of Theorem 1. It also demonstrates that positive examples can, in some cases, be much more useful and informative than negative examples. Sharma and Bilgic (2013) discuss the sources of uncertainty and reinforce our argument about the ineffectiveness of naive uncertainty sampling for some kinds of logical constraints.

We would like to emphasize the relationship between using Equation 2 as the scoring function, as proposed in Theorem 1, and using entropy (i.e., Equation 1) as the scoring function, as is done in uncertainty guided sampling. The following proposition and corollary of our theorem describe this relationship more precisely.

Proposition 1. *When:*

$$\underset{\substack{k=1,\ldots,K,\\ i=1,\ldots,N}}{\arg\max}\, p_k^i = \underset{\substack{k=1,\ldots,K,\\ i=1,\ldots,N}}{\arg\min}\, |p_k^i - 0.5|, \tag{3}$$

the probability scoring function of Equation 2 is equivalent to the entropy scoring function of Equation 1, which is used by uncertainty guided sampling.

Proof. The proof follows immediately by noticing that for $p_k^i \in [0,1]$, the following holds:

$$\underset{\substack{k=1,\ldots,K,\\ i=1,\ldots,N}}{\arg\max}\, \mathcal{S}_{\text{entropy}}(Y_k^i) \equiv \underset{\substack{k=1,\ldots,K,\\ i=1,\ldots,N}}{\arg\min}\, |p_k^i - 0.5|. \qquad \square$$

Corollary 1. *In the case of a single instance X the probability scoring function of Equation 2 and the entropy scoring function of Equation 1 are equivalent.*

Proof. Due to the mutual exclusion constraint, $\sum_{k=1}^{K} p_k \leq 1$, which implies that the condition of Proposition 1 is always satisfied. □

It is easy to observe that when we have several instances $X^1, \ldots, X^N$ and we compare the scores of each label-instance pair, then the two scoring functions are no longer necessarily equivalent. Also note that as the number of labels grows, the marginals are more likely to have smaller magnitudes and thus the condition of Proposition 1 is more likely to be satisfied.

A Different Approach. We now introduce a new concept that, when combined with the earlier motivation, gives rise to a new scoring function. The key intuition lies in the scenario where the discovery of a label being positive implies that all other labels are negative. This discovery may not be as valuable if the negative labels were already inferred to have low marginal probability. Instead, we propose to consider the *degree of surprise* of discovering that those labels are negative. For a label with marginal probability of being positive p_k, the amount of surprise can be defined in several ways. A function $\mathscr{S} : [0, 1] \mapsto \mathbb{R}$ is called a *surprise function* if it is decreasing and $\mathscr{S}(1) = 0$. A couple examples of such *surprise functions* are shown here:

- LOGARITHMIC: $\mathscr{S}_{\log}(p_k) \triangleq -\log p_k$. This is equivalent to the *self-information* of the event $Y_k = 1$ and was first referred to as a surprise measure by Tribus (1961).
- LINEAR: $\mathscr{S}_{\text{lin}}(p_k) \triangleq 1 - p_k$. This is associated to the 0-1 loss of Roy and McCallum (2001).

Using this definition of a surprise function, we define a new scoring function for the mutual exclusion case as follows:

$$\mathcal{S}_{\text{ME}}(Y_k) \triangleq p_k \underbrace{\sum_{c=1}^{K} \mathbb{1}_{c=k}\mathscr{S}(p_c) + \mathbb{1}_{c\neq k}\mathscr{S}(1-p_c)}_{\text{Total surprise of setting } Y_k=1}, \\ + (1-p_k)\underbrace{\mathscr{S}(1-p_k)}_{\text{Total surprise of setting } Y_k=0} \tag{4}$$

where $\mathscr{S}(\cdot)$ is an arbitrary surprise function, and $\mathbb{1}_{\cdot}$ is the indicator function which is equal to 1 if the condition in the subscript is satisfied and is equal to 0 otherwise. Note that the first term is the product of the probability of Y_k being equal to 1 and the sum of surprise "experienced" by fixing the value of Y_k to 1 (i.e., after propagating the mutual exclusion constraint, we sum over the surprises of all other labels being set to 0 and Y_k being set to 1). The second term is similarly defined as the product of the probability of Y_k being equal to 0 and the surprise of fixing Y_k to that value. No other variables are considered in this surprise value, as no other label value is fixed because of the mutual exclusion constraint. Note that this is substantially different than the entropy scoring function in that it's measuring "surprise" rather than uncertainty.

3.2. More General Logical Constraints

The scoring function of Equation 4 and the underlying intuition can easily be extended to more general logical constraints than mutual exclusion. An example of a more general logical constraint is *subsumption*. In this case, each label can have a set of parent and child labels, and a label being set to 1 implies that its parent label is 1. To extend the method introduced in the previous section, we need a function for propagating a fixed label-value pair through the constraints. Let this function be defined as $\mathcal{F}(Y_k = v) \triangleq \{(Y_{c_i}, v_i) : \text{if } Y_k = v, \text{ then } Y_{c_i} = v_i\}$, where $c_i \in \{1, \ldots, K\}$ is a label index, and $v_i \in \{0, 1\}$ is the value of Y_{c_i} fixed by propagating the fixed label-value pair (Y_k, v) through the constraints. We can now define our scoring function for general logical constraints as follows:

$$\mathcal{S}_{\text{constraints}}(Y_k) \triangleq p_k \underbrace{\sum_{(Y_{c_i}, v_i)\in\mathcal{F}(Y_k=1)} S(Y_{c_i}, v_i)}_{\text{Total surprise of setting } Y_k=1}, \\ + (1-p_k)\underbrace{\sum_{(Y_{c_i}, v_i)\in\mathcal{F}(Y_k=0)} S(Y_{c_i}, v_i)}_{\text{Total surprise of setting } Y_k=0}, \tag{5}$$

where $S(Y_{c_i}, v_i) = \mathbb{1}_{v_i=1}\mathscr{S}(p_{c_i}) + \mathbb{1}_{v_i=0}\mathscr{S}(1-p_{c_i})$.

Formal Justification. We have not derived a result for the general scoring function similar to that of Theorem 1. However, we can use the information-theoretic information gain to generate an interesting result, akin to our justification for using the scoring function of Equation 2 in the setting of mutual exclusion. We note that the information gain for the case with general logical constraints can be defined as a sum. The first term of this sum is the entropy of the label whose information gain is being computed. When the logarithmic surprise function is used with the scoring function of Equation 5, then our scoring function contains this entropy term, as well as an approximation of some other terms (but not all) of the complete information gain sum. More specifically, we have that (in this derivation we ignore terms that are constant across all label variables, since these terms do not affect the ranking of the labels induced by the information gain):

$$\begin{aligned}\mathcal{I}(Y_k) &= \mathcal{H}(Y_{-k}) - \mathcal{H}(Y_{-k} \mid \mathcal{H}_k), \\ &= \mathcal{H}(Y_{-k}) + \mathcal{H}(Y_k) - H(Y) = \mathcal{H}(Y_k) - \mathcal{H}(Y_k \mid \boldsymbol{Y_{-k}}), \\ &= \sum_{y_k} \Big[- \mathbb{P}(y_k) \log \mathbb{P}(y_k),\end{aligned}$$

$$
\begin{aligned}
&\quad + \sum_{\boldsymbol{y_{-k}}} \mathbb{P}(\boldsymbol{y_{-k}})\mathbb{P}(y_k \mid \boldsymbol{y_{-k}}) \log \mathbb{P}(y_k \mid \boldsymbol{y_{-k}}) \Big], \\
&= \sum_{y_k} \Big[- \mathbb{P}(y_k) \log \mathbb{P}(y_k), \\
&\qquad + \sum_{\boldsymbol{y_{-k}}} \big[\underbrace{\mathbb{P}(y_k, \boldsymbol{y_{-k}}) \log \mathbb{P}(y_k, \boldsymbol{y_{-k}})}_{\text{Constant}}, \\
&\qquad\qquad - \mathbb{P}(y_k, \boldsymbol{y_{-k}}) \log \mathbb{P}(\boldsymbol{y_{-k}}) \big] \Big], \\
&= \sum_{y_k} \mathbb{P}(y_k) \Big[- \log \mathbb{P}(y_k), \\
&\qquad - \sum_{\substack{\boldsymbol{y_{-f}} \\ \text{where } \boldsymbol{y_f} = \mathcal{F}(y_k)}} \underbrace{\mathbb{P}(\boldsymbol{y_{-f}}, \boldsymbol{y_{f \setminus k}} \mid y_k)}_{\mathbb{P}(\boldsymbol{y_{-f}} \mid \boldsymbol{y_f})} \log \mathbb{P}(\boldsymbol{y_{-f}}, \boldsymbol{y_{f \setminus k}}) \Big], \\
&= \sum_{y_k} \mathbb{P}(y_k) \Big[- \log \mathbb{P}(y_k), \\
&\qquad - \sum_{\substack{\boldsymbol{y_{-f}} \\ \text{where } \boldsymbol{y_f} = \mathcal{F}(y_k)}} \big[\underbrace{\mathbb{P}(\boldsymbol{y_{-f}} \mid \boldsymbol{y_f})}_{\text{Sums to 1}} \log \mathbb{P}(\boldsymbol{y_f}), \\
&\qquad\qquad + \mathbb{P}(\boldsymbol{y_{-f}} \mid \boldsymbol{y_f}) \log \mathbb{P}(\boldsymbol{y_{-f}} \mid \boldsymbol{y_{f \setminus k}}) \big] \Big], \\
&= \sum_{y_k} \mathbb{P}(y_k) \Big[- \underbrace{\log \mathbb{P}(y_k)}_{\text{Entropy}} - \underbrace{\log \mathbb{P}(\boldsymbol{y_f})}_{\text{Constraints}}, \\
&\qquad - \sum_{\substack{\boldsymbol{y_{-f}} \\ \text{where } \boldsymbol{y_f} = \mathcal{F}(y_k)}} \underbrace{\mathbb{P}(\boldsymbol{y_{-f}} \mid \boldsymbol{y_f}) \log \mathbb{P}(\boldsymbol{y_{-f}} \mid \boldsymbol{y_{f \setminus k}})}_{\text{Remainder}} \Big],
\end{aligned}
$$

where $\boldsymbol{f} \setminus k$ is the set of label indices in $\boldsymbol{f}$ excluding k. Note that the entropy scoring function of Equation 1 only considers the term denoted by "Entropy" in this sum. When the logarithmic surprise function is used, the general scoring function of Equation 5 contains an approximation to the terms denoted by "Constraints" where the joint is written as the product of the marginals. This result shows that the general scoring function is, in some sense, a better heuristic for the information gain than the entropy scoring function.

3.3. Computational Complexity

We will now consider the real-world use of an active learning system, where a request is made for a label of a particular instance. Note that if we were to use the information-theoretic information gain as our scoring function, then the cost would be linear in N and exponential in K. Our scoring functions reduce this cost. The entropy scoring function of Equation 1 has a computational cost linear in the number of labels and the number of instances (i.e., because we need to compute it for all labels); its cost is $O(NK)$. The probability scoring function of Equation 2 has the same cost. The mutual exclusion scoring function has cost $O(NK^2)$. Finally, ignoring the cost of the constraint propagation function, the general scoring function of Equation 5 has a computational cost of $O(NK^2)$, since the highest number of labels that can be fixed is K. Note that the constraint propagation function can have a cost exponential in K in the worst case. However, there are special cases where the cost of that operator is not as high. For example, with either the mutual exclusion or the subsumption constraint the cost is linear in K. When mutual exclusion is combined with subsumption, we can alternate between all our constraints, one by one, and keep propagating them, until no fixed label can be propagated further. If the number of constraints is C, the cost of that propagation operation is $O(CK)$. This is the most complex scenario that we consider in our experiments and covers most of the practical use cases in multi-task applications.

4. Experiments

In the following paragraphs, we describe the setup of our experiments, including the datasets and the evaluation metrics that we use, and the results and their corresponding analyses. All the datasets and code for the experiments are available at `https://github.com/eaplatanios/makina`.

We first define the names that we use to refer to different methods when plotting the results:

- `RANDOM` uses a random scoring function (i.e., using a random between 0 and 1 for the score).
- `ENTROPY` uses the entropy scoring function of Equation 1.
- `RANDOM-CP` is the same as `RANDOM`, but also propagates labels through the constraints.
- `ENTROPY-CP` is the same as `ENTROPY`, but also propagates labels through the constraints.
- `PROBABILITY-CP`: Using the probability scoring function of Equation 2 and also propagating labels through the constraints.
- `LOG-CP`: Using the constraints scoring function of Equation 5 with the logarithm surprise function and also propagating labels through the constraints.
- `LINEAR-CP`: Same as `LOG-CP`, but using the linear surprise function instead of the logarithm.

We apply the same experimental setup to all of the datasets. Each dataset consists of a set of positive examples for each label. For each experiment, we split the dataset into training and testing subsets. For each label, we train a binary logistic regression classifier using the AdaGrad stochastic optimization algorithm of (Duchi et al., 2011), with a batch size of 100 samples per iteration. Our experimental pipeline consists of the following steps:

1. We initially train a classifier for each label independently using the training portion of the dataset. We consider all positive examples for the corresponding label, along with a set of negative examples of the same size, sampled from the remaining set of examples in the training dataset.
2. We repeat the following steps until all of the testing data have been manually labeled:
   (a) Request a set of M examples from the testing dataset to be manually labeled[2], sequentially. For

[2]Note that by "example" we mean a label-instance pair and

Table 1: Datasets used in experiments.

DATASET	#CLASSES	#FEATURES	BALANCED?	#TRAINING	#TESTING	#REQUESTED/ITERATION
SATIMAGE	6	36	×	3,104	1,331	100
SHUTTLE	7	9	×	30,450	13,050	1,000
SEGMENT	7	19	√	400	1,910	100
PENDIGITS	10	16	√	7,494	3,498	100
LETTER	26	16	√	15,000	5,000	1,000
NELL-7	7	180,878	×	214	14,693	500
NELL-11	11	180,878	×	242	14,693	1,000
NELL-13	13	180,878	×	2,656	18,016	2,000

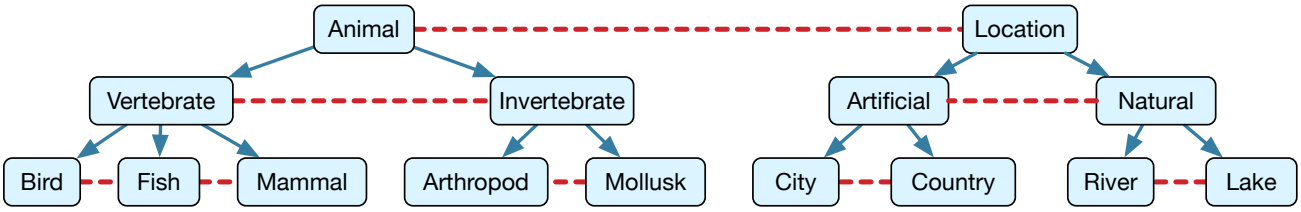

Figure 2: Illustration of the NELL-13 dataset constraints. Each box represents a label, each blue arrow represents a subsumption constraint, and each set of labels connected by a red dashed line represents a mutually exclusive set of labels.

all the methods that include CP in their name, after each example is obtained, propagate all the logical constraints. The examples fixed by this process are considered manually labeled. Note that M can vary across each dataset since they differ in size. Please refer to Table 1 for the values of M used for each dataset. Also note that this step differs across our methods. Each method's scoring function determines which examples are selected for labeling. The label-instance pair with the highest score is selected for labeling.

(b) Move all the labeled examples from the testing to the training dataset.

(c) Re-train the classifiers for all the labels, using the updated training dataset. Training for the classifiers is initialized at the previously learned point to reduce convergence time.

(d) Evaluate progress using a set of metrics and the full dataset (i.e., the training and testing parts of the dataset, combined). Note that even though it may seem unorthodox to evaluate on the full dataset, it is actually meaningful for settings like NELL. In fact, that is how NELL is evaluated, as we care about the accuracy of its whole knowledge-base, irrespective of how the label of an instance was obtained.

A Note on Marginal Probabilities. Note that all our methods and results of section 3 rely on marginal probabilities. In our experiments, we use classifiers to estimate those marginals and sometimes they may not be very accurate. This is actually the reason we subsample a number of negative examples equal to the number of positive examples. Otherwise, our logistic regression classifiers would be biased towards low estimates of the probabilities, which would cause the entropy and our proposed scoring functions to perform very similarly, as shown in section 3.1. This was indeed the case when we ran experiments without subsampling the negative examples. This problem can also be alleviated by using more appropriate classifiers for the problem, than logistic regression.

so all possible label instance pairs from the testing dataset are considered at this stage.

4.1. Datasets

We now provide the list of data sets we used for our experiments, with a small description for each data set. Table 1 provides details on the statistics and experimental setup for each dataset. All data sets, except for the NELL data, were obtained from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html`. The NELL data sets were obtained from `https://rtw.ml.cmu.edu/rtw/resources`. Details on the statistics and experimental setup for each data set are provided in table 1. Note that for all data sets except NELL-11 and NELL-13, the only constraint used is a mutual exclusion constraint between all labels. The constraints used for those two NELL data sets are detailed in the following list.

- SATIMAGE: Classify a satellite image region (Feng et al., 1993).
- SHUTTLE: Classify a space shuttle as belonging to one of seven classes (Feng et al., 1993).
- SEGMENT: Classify a small outdoor image region (Feng et al., 1993).
- PENDIGITS: Classify a handwritten digit (Alimoglu and Alpaydin, 1996).
- LETTER: Classify an image as a letter of the English alphabet (Frey and Slate, 1991).

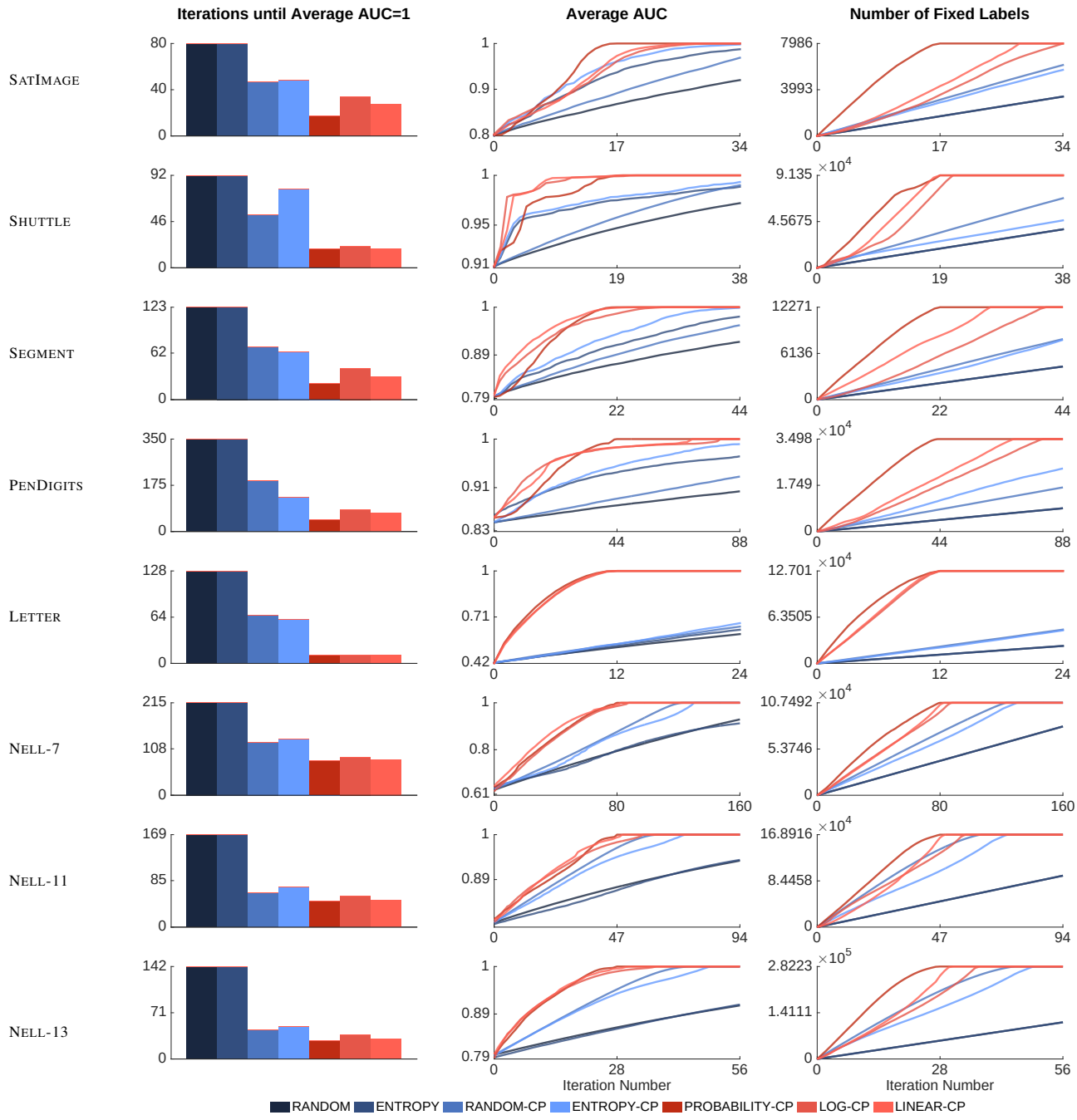

Figure 3: Results. The red colored plots refer to the proposed methods and the blue colored plots refer to existing methods (apart from the constraint propagation step that we optionally added to all existing methods to enable a fairer comparison, and which is denoted with a `-CP` appended to the method name). For the first plot, the lower the bar, the better the result. For the rest of the plots, the higher the value of the curve per iteration, the better the result. We thus observe that the proposed methods outperform all existing methods for all of the experiments performed.

- NELL-7: Classify noun phrases as belonging to a certain category or not. The categories considered for this data set are `Bird`, `Fish`, `Mammal`, `City`, `Country`, `Lake`, and `River` (i.e., the category represents the label in this case). The only constraint considered in this case is that all these categories are mutually exclusive. We use the same set of features as that used by the coupled pattern learner (CPL) in NELL (Mitchell et al., 2015).

- NELL-11: Perform the same task as NELL-7, but additionally consider the categories `Animal`, `Location`, `Artificial Location`, and `Natural Location`. Also include the subsumption constraints shown in figure 2, while ignoring the categories not included in this data set.
- NELL-13: Perform the same task as NELL-7, but with the categories and constraints illustrated in figure 2.

4.2. Evaluation Metrics

We first define *average area under the curve (average AUC)*. At each iteration and for each label, we compute the AUC over the whole dataset (i.e., training dataset and testing dataset combined). Then, we compute a weighted average of the AUCs for each label, where each label's contribution is weighted by the number of positive examples that exist in the dataset, for that label. That weighted average is what we refer to as average AUC. It is easy to see that average AUC is a non-decreasing function with respect to iteration number[3]. We use the following three metrics to evaluate the proposed methods:

- Iterations until Average AUC=1: Number of iterations until average AUC ≥ 0.999.
- Average AUC: Average AUC vs iteration number.
- Number of Fixed Labels: Number of labels that are effectively fixed (i.e., added to the training dataset), after each iteration. Note that this measure is not always equal to the number of labels requested because of the constraint propagation step.

4.3. Results Analysis

All results are shown in Figure 3. We first note that the proposed methods *consistently beat the other methods by a significant margin, for all datasets and all evaluation metrics*. For the datasets that only consider a single mutual exclusion constraint, `PROBABILITY-CP` always performs best with respect to the number of iterations until average AUC=1. This is not unexpected; as we showed in Section 3, this method can be considered optimal. Furthermore, we find it interesting that, for the average AUC plots, in all cases where we only have a single mutual exclusion constraint, despite seeing underperformance in early iterations, `PROBABILITY-CP` still reaches AUC $= 1$ faster. This may be based on the fact that this method first selects label-instance pairs with probability very close to 1, which turn out to be positive. However, after a few iterations, the method experiences a boost and beats all of the other methods. As for the number of fixed labels per iteration, `PROBABILITY-CP` also beats the other methods by far. This provides validation of the intuition discussed in Section 3, that the method would fix more labels when the mutual exclusion constraint is propagated. As for the two datasets where we also have subsumption constraints and multiple mutual exclusion constraints, we see that the proposed methods consistently outperform all other methods, as expected. We did not expect, however, for `PROBABILITY-CP` to be doing as well as `LOG-CP` and `LINEAR-CP`. We do not yet have an understanding about this finding, but we find this interesting and encouraging for the proposed methods. We note that there are a few datasets with only a single mutual exclusion constraint, where `PROBABILITY-CP` is actually beaten early on in the average AUC curve, by our other two proposed methods. Thus, there is value in using these two methods in some scenarios. Finally, we found interesting that the constraint propagation step alone provides a significant performance boost to all methods.

[3]That nice property is the reason we use the combined dataset as opposed to just using the testing dataset.

5. Conclusion

We have proposed methods for performing active learning efficiently in the presence of logical constraints between the outputs of multiple classifiers. The approach resonates with underlying intuitions and challenges the core idea behind uncertainty guided sampling. We provided theoretical justification for using the proposed methods. In a set of experiments, we found that the methods consistently outperformed competing methods across ten diverse datasets and thus appear to be promising for practical applications. Moreover, the experiments showed that our methods can be used to speed up the learning process in NELL. Per our knowledge, this paper is the first to describe and carefully study methods for performing active learning when there are logical constraints among outputs of multiple classifiers.

We are excited about numerous future directions for this work. Our first priority is to pursue additional theoretical results for the general setting with arbitrary logical constraints. We would also like to explore methods for a setting where all labels for a particular data instance are requested at each iteration; this use case is useful to systems like NELL where the label space is extremely sparse. We would also like to explore ways in which we can use accuracy estimates for the trained classifiers (using methods such as those proposed in (Platanios et al., 2016) and (Platanios et al., 2017), which uses similar logical constraints, for example) in order to make the active learning procedures more robust. Implementing efficient computation of the value of information for multiple, interdependent classifiers would be a step towards autonomous learning systems with the ability to reflect more deeply about their pursuit of information.

Acknowledgements

We would like to thank Abulhair Saparov and Otilia Stretcu for the useful feedback they provided in early versions of this paper. This research was performed during an internship at Microsoft Research.

References

F. Alimoglu and E. Alpaydin. Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96*, 1996.

G. Angeli, J. Tibshirani, J. Y. Wu, and C. D. Manning. Combining Distant and Partial Supervision for Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1556–1567, 2014.

M. Bilgic, L. Mihalkova, and L. Getoor. Active Learning for Networked Data. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.

M. Chang, L. Ratinov, and D. Roth. Guiding semi-supervision with constraint-driven learning. In *Annual Meeting of the Association of Computational Linguistics*, 2007.

M. Chang, L. Ratinov, N. Rizzolo, and D. Roth. Learning and inference with constraints. In *National Conference on Artificial Intelligence*, pages 1513–1518, 2008.

A. Culotta and A. McCallum. Reducing Labeling Effort for Structured Prediction Tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI '05, pages 746–751, 2005.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

C. Feng, A. Sutherland, R. King, S. Muggleton, and R. Henery. Comparison of machine learning classifiers to statistics and neural networks. In *Proceedings of the Third International Workshop in Artificial Intelligence and Statistics*, pages 41–52, 1993.

P. W. Frey and D. J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6:161, 1991.

A. Harpale. *Multi-Task Active Learning*. PhD thesis, 2012.

A. Kapoor and S. Baker. Which faces to tag: Adding prior constraints into active learning. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, September 2009.

A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *International Joint Conference on Artificial Intelligence*, pages 877–882, 2007.

A. Krause and C. Guestrin. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research*, 35:557–591, 2009.

W. Luo, A. Schwing, and R. Urtasun. Latent Structured Active Learning. In *Advances in Neural Information Processing Systems 26*, pages 728–736. 2013.

T. M. Mitchell, W. W. Cohen, E. R. Hruschka Jr, P. Pratim Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. P. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. C. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. In *Association for the Advancement of Artificial Intelligence*, pages 1–9, 2015.

E. A. Platanios, A. Dubey, and T. M. Mitchell. Estimating Accuracy from Unlabeled Data: A Bayesian Approach. In *International Conference in Machine Learning*, pages 1416–1425, 2016.

E. A. Platanios, H. Poon, E. Horvitz, and T. M. Mitchell. Estimating Accuracy from Unlabeled Data: A Probabilistic Logic Approach. In *Neural Information Processing Systems*, 2017.

K. L. Poh and E. Horvitz. A Graph-Theoretic Analysis of Information Value. In *Conference on Uncertainty in Artificial Intelligence*, pages 1–10, 1996.

R. Reichart, K. Tomanek, U. Hahn, and A. Rappoport. Multi-task active learning for linguistic annotations. In *Annual Meeting of the Association for Computational Linguistics*, pages 861–869, 2008.

D. Roth and K. Small. Active Learning for Pipeline Models. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2008.

N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*, pages 441–448, 2001.

P. Ruvolo and E. Eaton. Active Task Selection for Lifelong Machine Learning. In *Association for the Advancement of Artificial Intelligence*, 2013.

B. Settles. *Active Learning*, volume 6. Morgan & Claypool Publishers, June 2012.

B. Settles and M. Craven. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1070–1079, 2008.

M. Sharma and M. Bilgic. Most-Surely vs. Least-Surely Uncertain. In *IEEE International Conference on Data Mining, ICDM*, 2013.

M. Tribus. *Thermostatics and Thermodynamics*. D. Van Nostrand Company, 1961.

B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. Active Learning for Biomedical Citation Screening. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 173–182, 2010.

Y. Zhang. Multi-task active learning with output constraints. In *Association for the Advancement of Artificial Intelligence*, 2010.

S. Zhao, J. Wu, V. S. Sheng, C. Ye, P. Zhao, and Z. Cui. Weak labeled multi-label active learning for image classification. In *ACM International Conference on Multimedia*, pages 1127–1130, 2015.