Learning and Inference over Constrained Output

Vasin Punyakanok **Dan Roth** **Wen-tau Yih** **Dav Zimak**
Department of Computer Science
University of Illinois at Urbana-Champaign
{punyakan, danr, yih, davzimak}@uiuc.edu

Abstract

We study learning structured output in a discriminative framework where values of the output variables are estimated by local classifiers. In this framework, complex dependencies among the output variables are captured by constraints and dictate which global labels can be inferred. We compare two strategies, *learning independent classifiers* and *inference based training*, by observing their behaviors in different conditions. Experiments and theoretical justification lead to the conclusion that using inference based learning is superior when the local classifiers are difficult to learn but may require many examples before any discernible difference can be observed.

1 Introduction

Making decisions in real world problems involves assigning values to sets of variables where a complex and expressive structure can influence, or even dictate, what assignments are possible. For example, in the task of identifying named entities in a sentence, prediction is governed by constraints like "entities do not overlap." Another example exists in scene interpretation tasks where predictions must respect constraints that could arise from the nature of the data or task, such as "humans have two arms, two legs, and one head."

There exist at least three fundamentally different solutions to learning classifiers over structured output. In the first, structure is ignored; local classifiers are learned and used to predict each output component separately. In the second, learning is decoupled from the task of maintaining structured output. Estimators are used to produce global output consistent with the structural constraints only after they are learned for each output variable separately. Discriminative HMM, conditional models [Punyakanok and Roth, 2001; McCallum *et al.*, 2000] and many dynamic programming based schemes used in the context of sequential predictions fall into the this category. The third class of solutions incorporates dependencies among the variables into the learning process to directly induce estimators that optimize a global performance measure. Traditionally these solutions were generative; however recent developments have produced discriminative models of this type, including conditional random fields [Lafferty *et al.*, 2001], Perceptron-based learning of structured output [Collins, 2002; Carreras and Màrquez, 2003] and Max-Margin Markov networks which allow incorporating Markovian assumptions among output variables [Taskar *et al.*, 2004].

Incorporating constraints during training can lead to solutions that directly optimize the true objective function, and hence, should perform better. Nonetheless, most real world applications using this technique do not show significant advantages, if any. Therefore, it is important to discover the tradeoffs of using each of the above schemes.

In this paper, we compare three learning schemes. In the first, classifiers are learned independently (*learning only* (LO)), in the second, inference is used to maintain structural consistency only after learning (*learning plus inference* (L+I)), and finally inference is used while learning the parameters of the classifier (*inference based training* (IBT)). In semantic role labeling (SRL), it was observed [Punyakanok *et al.*, 2004; Carreras and Màrquez, 2003] that when the local classification problems are easy to learn, L+I outperforms IBT. However, when using a reduced feature space where the problem was no longer (locally) separable, IBT could overcome the poor local classifications to yield accurate global classifications.

Section 2 provides the formal definition of our problem. For example, in Section 3, we compare the three learning schemes using the online Perceptron algorithm applied in the three settings (see [Collins, 2002] for details). All three settings use the same linear representation, and L+I and IBT share the same decision function space. Our conjectures of the relative performance between different schemes are presented in Section 4. Despite the fact that IBT is a more powerful technique, in Section 5, we provide an experiment that shows how L+I can outperform IBT when there exist accurate local classifiers that do not depend on structure, or when there are too few examples to learn complex structural dependencies. This is also theoretically justified in Section 6.

2 Background

Structured output classification problems have many flavors. In this paper, we focus on problems where it is natural both to split the task into many smaller classification tasks and to solve directly as a single task. In Section 5.2, we consider the

semantic role-labeling problem, where the input $\mathcal{X}$ are natural language features and the output $\mathcal{Y}$ is the position and type of a semantic-role in the sentence. For this problem, one can either learn a set of local functions such as "is this phrase an argument of 'run'," or a global classifier to predict all semantic-roles at once. In addition, natural structural constraints dictate, for example, that no two semantic roles for a single verb can overlap. Other structural constraints, as well as linguistic constraints yield a restricted output space in which the classifiers operate.

In general, given an assignment $\mathbf{x} \in \mathcal{X}^{n_x}$ to a collection of input variables, $\mathbf{X} = (X_1, \ldots, X_{n_x})$, the structured classification problem involves identifying the "best" assignment $\mathbf{y} \in \mathcal{Y}^{n_y}$ to a collection of output variables $\mathbf{Y} = (Y_1, \ldots, Y_{n_y})$ that are consistent with a defined structure on $\mathbf{Y}$. This structure can be thought of as constraining the output space to a smaller space $\mathcal{C}(\mathcal{Y}^{n_y}) \subseteq \mathcal{Y}^{n_y}$, where $\mathcal{C} : 2^{\mathcal{Y}^*} \rightarrow 2^{\mathcal{Y}^*}$ constrains the output space to be structurally consistent.

In this paper, a structured output classifier is a function $h : \mathcal{X}^{n_x} \rightarrow \mathcal{Y}^{n_y}$, that uses a global scoring function, $f : \mathcal{X}^{n_x} \times \mathcal{Y}^{n_y} \rightarrow \mathbb{R}$ to assign scores to each possible example/label pair. Given input $\mathbf{x}$, it is hoped that the correct output $\mathbf{y}$ achieves the highest score among consistent outputs:

$$\hat{\mathbf{y}} = h(\mathbf{x}) = \underset{\mathbf{y}' \in \mathcal{C}(\mathcal{Y}^{n_y})}{\operatorname{argmax}} f(\mathbf{x}, \mathbf{y}'), \tag{1}$$

where n_x and n_y depend on the example at hand. In addition, we view the global scoring function as a composition of a set of local scoring functions $\{f_y(\mathbf{x}, t)\}_{y \in \mathcal{Y}}$, where $f_y : \mathcal{X}^{n_x} \times \{1, \ldots, n_y\} \rightarrow \mathbb{R}$. Each function represents the *score* or confidence that output variable Y_t takes value y:

$$f(\mathbf{x}, (y_1, \ldots, y_{n_y})) = \sum_{t=1}^{n_y} f_{y_t}(\mathbf{x}, t)$$

Inference is the task of determining an optimal assignment $\mathbf{y}$ given an assignment $\mathbf{x}$. For sequential structure of constraints, polynomial-time algorithms such as Viterbi or CSCL [Punyakanok and Roth, 2001] are typically used for efficient inference. For general structure of constraints, a generic search method (e.g., beam search) may be applied. Recently, integer programming has also been shown to be an effective inference approach in several NLP applications [Roth and Yih, 2004; Punyakanok *et al.*, 2004].

In this paper, we consider classifiers with *linear representation*. Linear local classifiers are linear functions, $f_y(\mathbf{x}, t) = \boldsymbol{\alpha}^y \cdot \Phi^y(\mathbf{x}, t)$, where $\boldsymbol{\alpha}^y \in \mathbb{R}^{d_y}$ is a weight vector and $\Phi^y(\mathbf{x}, t) \in \mathbb{R}^{d_y}$ is a feature vector. Then, it is easy to show that the global scoring function can be written in the familiar form $f(\mathbf{x}, \mathbf{y}) = \boldsymbol{\alpha} \cdot \Phi(\mathbf{x}, \mathbf{y})$, where $\Phi^y(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n_y} \Phi^{y_t}(\mathbf{x}, t) I_{\{y_t = y\}}$ is an accumulation over all output variables of features occurring for class y, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^{|\mathcal{Y}|})$ is concatenation of the $\boldsymbol{\alpha}^y$'s, and $\Phi(\mathbf{x}, \mathbf{y}) = (\Phi^1(\mathbf{x}, \mathbf{y}), \ldots, \Phi^{|\mathcal{Y}|}(\mathbf{x}, \mathbf{y}))$ is the concatenation of the $\Phi^y(\mathbf{x}, \mathbf{y})$'s. Then, the global classifier is

$$h(\mathbf{x}) = \hat{\mathbf{y}} = \underset{\mathbf{y}' \in \mathcal{C}(\mathcal{Y}^{n_y})}{\operatorname{argmax}} \boldsymbol{\alpha} \cdot \Phi(\mathbf{x}, \mathbf{y}').$$

3 Learning

We present several ways to learn the scoring function parameters differing in whether or not the structure-based inference process is leveraged during training. Learning consists of choosing a function $h : \mathcal{X}^* \rightarrow \mathcal{Y}^*$ from some hypothesis space, $\mathcal{H}$. Typically, the data is supplied as a set $\mathbf{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$ from a distribution $\mathrm{P}_{\mathcal{X},\mathcal{Y}}$ over $\mathcal{X}^* \times \mathcal{Y}^*$. While these concepts are very general, we focus on online learning of linear representations using a variant of the Perceptron algorithm (see [Collins, 2002]).

Learning Local Classifiers: Learning stand-alone local classifiers is perhaps the most straightforward setting. No knowledge of the inference procedure is used. Rather, for each example $(\mathbf{x}, \mathbf{y}) \in \mathbf{D}$, the learning algorithm must ensure that $f_{y_t}(\mathbf{x}, t) > f_{y'}(\mathbf{x}, t)$ for all $t = 1, \ldots, n_y$ and all $y' \neq y_t$. In Figure 3(a), an online Perceptron-style algorithm is presented where no global constraints are used. See [Har-Peled *et al.*, 2003] for details and Section 5 for experiments.

Learning Global Classifiers: We seek to train classifiers so they will produce the correct *global* classification. To this end, the key difference from learning locally is that feedback from the inference process determines which classifiers to modify so that together, the classifiers and the inference procedure yield the desired result. As in [Collins, 2002; Carreras and Màrquez, 2003], we train according to a global criterion. The algorithm presented here is an online procedure, where at each step a subset of the classifiers are updated according to inference feedback. See Figure 3(b) for details of a Perceptron-like algorithm for learning with inference feedback.

Note that in practice it is common for problems to be modeled in such a way that local classifiers are dependent on part of the output as part of their input. This sort of *interaction* can be incorporated directly to the algorithm for learning a global classifier as long as an appropriate inference process is used. In addition, to provide a fair comparison between LO, L+I, and IBP in this setting one must take care to ensure that the learning algorithms are appropriate for this task. In order to remain focused on the problem of training with and without inference feedback, the experiments and analysis presented concern only the local classifiers without interaction.

4 Conjectures

In this section, we investigate the relative performance of classifier systems learned with and without inference feedback. There are many competing factors. Initially, if the local classification problems are "easy", then it is likely that learning local classifiers only (LO) can yield the most accurate classifiers. However, an accurate model of the structural constraints could additionally increase performance (learning plus inference (L+I)). As the local problems become more difficult to learn, an accurate model of the structure becomes more important, and can, perhaps, overcome sub-optimal local classifiers. Despite the existence of a global solution, as the local classification problems become increasingly difficult, it is unlikely that structure based inference can fix poor classifiers learned locally. In this case, only training with inference feedback (IBT) can be expected to perform well.

Algorithm ONLINELOCALLEARNING
INPUT: $\mathbf{D}^{X,Y} \in \{\mathcal{X}^* \times \mathcal{Y}^*\}^m$
OUTPUT: $\{f_y\}_{y\in\mathcal{Y}} \in \mathcal{H}$

Initialize $\boldsymbol{\alpha}^y \in \mathbb{R}^{|\Phi^y|}$ for $y \in \mathcal{Y}$
Repeat until converge
for each $(\mathbf{x}, \mathbf{y}) \in \mathbf{D}^{X,Y}$ **do**
for $t = 1, \ldots, n_y$ **do**
$\hat{y}_t = \text{argmax}_y\, \boldsymbol{\alpha}^y \cdot \Phi^y(\mathbf{x}, t)$
if $\hat{y}_t \neq y_t$ **then**
$\boldsymbol{\alpha}^{y_t} = \boldsymbol{\alpha}^{y_t} + \Phi^{y_t}(\mathbf{x}, t)$
$\boldsymbol{\alpha}^{\hat{y}_t} = \boldsymbol{\alpha}^{\hat{y}_t} - \Phi^{\hat{y}_t}(\mathbf{x}, t)$

(a) Without inference feedback

Algorithm ONLINEGLOBALLEARNING
INPUT: $\mathbf{D}^{X,Y} \in \{\mathcal{X}^* \times \mathcal{Y}^*\}^m$
OUTPUT: $\{f_y\}_{y\in\mathcal{Y}} \in \mathcal{H}$

Initialize $\boldsymbol{\alpha} \in \mathbb{R}^{|\Phi|}$
Repeat until converge
for each $(\mathbf{x}, \mathbf{y}) \in \mathbf{D}^{X,Y}$ **do**
$\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}\in\mathcal{C}(\mathcal{Y}^{n_y})}\, \boldsymbol{\alpha} \cdot \Phi(\mathbf{x}, \mathbf{y})$
if $\hat{\mathbf{y}} \neq \mathbf{y}$ **then**
$\boldsymbol{\alpha} = \boldsymbol{\alpha} + \Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}})$

(b) With inference feedback

Figure 1: Algorithms for learning without and with inference feedback. The key difference lies in the inference step (i.e. argmax). Inference while learning locally is trivial and the prediction is made simply by considering each label locally. Learning globally uses a global inference (i.e. $\text{argmax}_{\mathbf{y}\in\mathcal{C}(\mathcal{Y}^{n_y})}$) to predict global labels.

As a first attempt to formalize the difficulty of classification tasks, we define separability and learnability. A classifier, $f \in \mathcal{H}$, *globally separates* a data set $\mathbf{D}$ iff for all examples $(\mathbf{x}, \mathbf{y}) \in \mathbf{D}$, $f(\mathbf{x}, \mathbf{y}) > f(\mathbf{x}, \mathbf{y}')$ for all $\mathbf{y}' \in \mathcal{Y}^{n_y} \setminus \mathbf{y}$ and *locally separates* $\mathbf{D}$ iff for all examples $(\mathbf{x}, \mathbf{y}) \in \mathbf{D}$, $f_{y_t}(\mathbf{x}, t) > f_y(\mathbf{x}, t)$ for all $y \in \mathcal{Y} \setminus y_t$, and all $\mathbf{y}' \in \mathcal{Y}^{n_y} \setminus \mathbf{y}$. A learning algorithm $\mathcal{A}$ is a function from data sets to a $\mathcal{H}$. We say that $\mathbf{D}$ is *globally (locally) learnable* by $\mathcal{A}$ if there exists an $f \in \mathcal{H}$ such that f *globally (locally) separates* $\mathbf{D}$.

The following simple relationships exist between local and global learning: 1. local separability implies global separability, but the inverse is not true; 2. local separability implies local and global learnability; 3. global separability implies global learnability, but not local learnability. As a result, it is clear that if there exist learning algorithms to learn global separations, then given enough examples, IBT will outperform L+I. However, learning examples are often limited either because they are expensive to label or because some learning algorithms simply do not scale well to many examples. With a fixed number of examples, L+I can outperform IBT.

Claim 4.1 *With a fixed number of examples:*

1. *If the local classification tasks are separable, then L+I outperforms IBT.*
2. *If the task is globally separable, but not locally separable then IBT outperforms L+I only with sufficient examples. This number correlates with the degree of the separability of the local classifiers.*

5 Experiments

We present experiments to show how the relative performance of learning plus inference (L+I) compares to inference based training (IBT) when the quality of the local classifiers and amount of training data varies.

5.1 Synthetic Data

In our experiment, each example $\mathbf{x}$ is a set of c points in d-dimensional real space, where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_c) \in \mathbb{R}^d \times \ldots \times \mathbb{R}^d$ and its label is a sequence of binary variable, $\mathbf{y} = (y_1, \ldots, y_c) \in \{0, 1\}^c$, labeled according to:

$$\mathbf{y} = h(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{C}(\mathcal{Y}^c)}{\text{argmax}} \sum_i y_i f_i(\mathbf{x}_i) - (1 - y_i) f_i(\mathbf{x}_i),$$

where $\mathcal{C}(\mathcal{Y}^c)$ is a subset of $\{0, 1\}^c$ imposing a random constraint[1] on $\mathbf{y}$, and $f_i(\mathbf{x}_i) = \mathbf{w}_i\mathbf{x}_i + \theta_i$. Each f_i corresponds to a local classifier $y_i = g_i(\mathbf{x}_i) = I_{f_i(\mathbf{x}_i)>0}$. Clearly, the dataset generated from this hypothesis is globally linearly separable. To vary the difficulty of local classification, we generate examples with various degree of linear separability of the local classifiers by controlling the fraction κ of the data where $h(\mathbf{x}) \neq g(\mathbf{x}) = (g_1(\mathbf{x}_1), \ldots, g_c(\mathbf{x}_c))$—examples whose labels, if generated by local classifiers independently, violate the constraints (i.e. $g(\mathbf{x}) \notin \mathcal{C}(\mathcal{Y}^c)$).

Figure 2 compares the performance of different learning strategies relative to the number of training examples used. In all experiments, $c = 5$, the true hypothesis is picked at random, and $\mathcal{C}(\mathcal{Y}^c)$ is a random subset with half of the size of $\mathcal{Y}^c$. Training is halted when a cycle complete with no errors, or 100 cycles is reached. The performance is averaged over 10 trials. Figure 2(a) shows the locally linearly separable case where L+I outperforms IBT. Figure 2(c) shows results for the case with the most difficult local classification tasks($\kappa = 1$) where IBT outperforms L+I. Figure 2(b) shows the case where data is not totally locally linearly separable($\kappa = 0.1$). In this case, L+I outperforms IBT when the number of training examples is small. In all cases, inference helps.

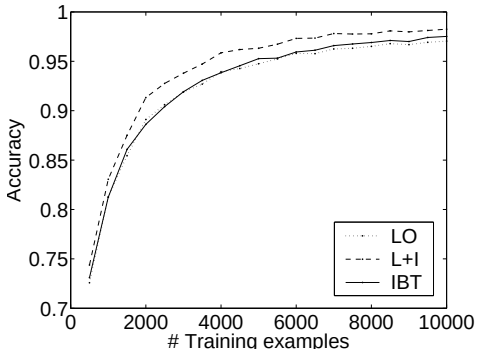
5.2 Real-World Data

In this section, we present experiments on two real-world problems from natural language processing – semantic role labeling and noun phrase identification.

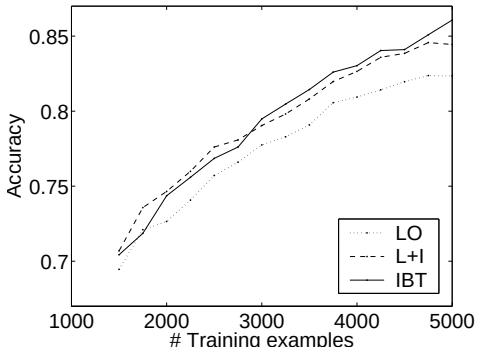
Semantic-Role Labeling

Semantic role labeling (SRL) is believed to be an important task toward natural language understanding, and has immediate applications in tasks such Information Extraction and

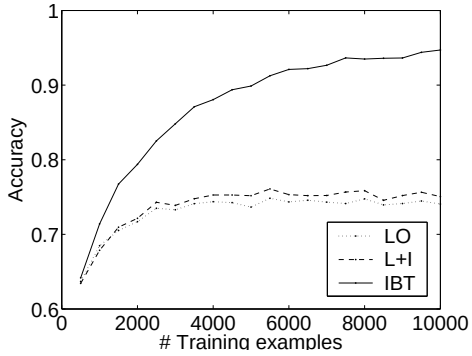
[1] Among the total 2^c possible output labels, $\mathcal{C}(\cdot)$ fixes a random fraction as legitimate global labels.



(a) $\kappa = 0, d = 100$



(b) $\kappa = 0.15, d = 300$



(c) $\kappa = 1, d = 100$

Figure 2: Comparison of different learning strategies in various degrees of difficulties of the local classifiers. $\kappa = 0$ implies locally linearly separability. Higher κ indicates harder local classification.

Question Answering. The goal is to identify, for each verb in the sentence, all the constituents which fill a semantic role, and determine their argument types, such as *Agent*, *Patient*, *Instrument*, as well as adjuncts such as *Locative*, *Temporal*, *Manner*, etc. For example, given a sentence " I *left* my pearls to my daughter-in-law in my will", the goal is to identify different arguments of the verb *left* which yields the output:

$[_{\text{A0}}$ I$]$ $[_{\text{V}}$ *left* $]$ $[_{\text{A1}}$ my pearls$]$ $[_{\text{A2}}$ to my daughter-in-law$]$ $[_{\text{AM-LOC}}$ in my will$]$.

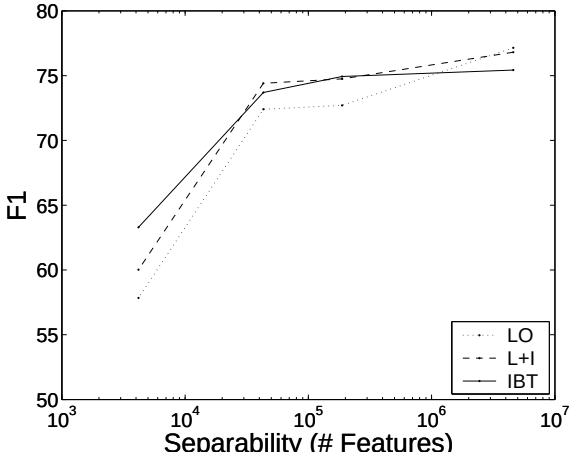


Figure 3: Results on the semantic-role labeling (SRL) problem. As the number of features increases, the difficulty of the local classification problem becomes easier, and the independently learned classifiers (LO) perform well, especially when inference is used after learning (L+I). Using inference during training (IBT) can aid performance when the learning problem is more difficult (few features).

Here A0 represents *leaver*, A1 represents *thing left*, A2 represents *benefactor*, AM-LOC is an adjunct indicating the location of the action, and V determines the verb.

We model the problem using classifiers that map constituent candidates to one of 45 different types, such as f_{AO} and f_{A1} [Kingsbury and Palmer, 2002; Carreras and Màrquez, 2003]. However, local multiclass decisions are insufficient. Structural constraints are necessary to ensure, for example, that no arguments can overlap or embed each other. In order to include both structural and linguistic constraints, we use a general inference procedure based on integer linear programming [Punyakanok *et al.*, 2004]. We use data provided in the CoNLL-2004 shared task [Carreras and Màrquez, 2003], but we restrict our focus to sentences that have greater than five arguments. In addition, to simplify the problem, we assume the boundaries of the constituents are given – the task is mainly to assign the argument types.

The experiments clearly show that IBT outperforms locally learned LO and L+I when the local classifiers are inseparable and difficult to learn. The difficulty of local learning was controlled by varying the number of input features. With more features, the linear classifier are more expressive and can learn effectively and L+I outperforms IBT. With less features the problem becomes more difficult and IBT outperforms L+I. See Figure 3.

Noun Phrase Labeling

Noun phrase identification involves the identification of phrases or of words that participate in a syntactic relationship. Specifically, we use the standard base Noun Phrases (NP) data set [Ramshaw and Marcus, 1995] taken from the Wall Street Journal corpus in the Penn Treebank [Marcus *et al.*, 1993].

The phrase identifier consists of two classifiers: one that

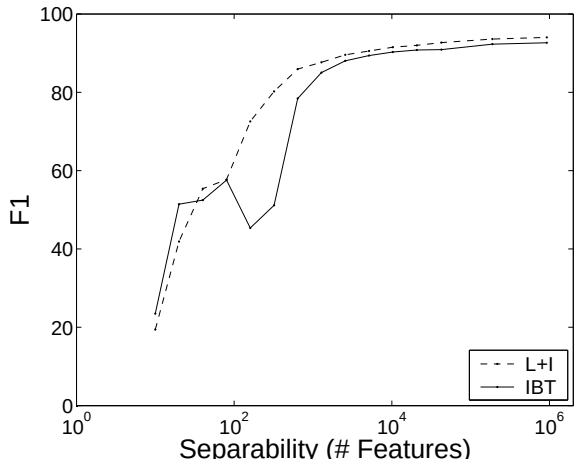


Figure 4: Results on the noun phrase (NP) identification problem.

detects the beginning, $f_{[}$, and a second that detects the end, $f_{]}$ of a phrase. The outcome of these classifiers are then combined in a way that satisfies structural constraints constraints (e.g. non-overlapping), using an efficient constraint satisfaction mechanism that makes use of the confidence in the classifiers' outcomes [Punyakanok and Roth, 2001].

In this case, L+I trains each classifier independently, and only during evaluation, the inference is used. On the other hand, IBT incorporates the inference into the training. For each sentence, each word position is processed by the classifiers, and their outcomes are used by the inference process to infer the final prediction. The classifiers are then updated based on the final prediction not on their own prediction before the inference.

As in the previous experiment, Figure 4 shows performance of two systems varied by the number of features. Unlike the previous experiment, the number of features in each experiment was determined by the frequency of occurrence. Less frequent features are pruned to make the task more difficult. The results are similar to the SRL task in that only when the problem becomes difficult IBT outperforms L+I.

6 Bound Prediction

In this section, we use standard VC-style generalization bounds from learning theory to gain intuition into when learning locally (LO and L+I) may outperform learning globally (IBT) by comparing the expressivity and complexity of each hypothesis space. When learning globally, it is possible to learn concepts that may be difficult to learn locally, since the global constraints are not available to the local algorithms. On the other hand, while the global hypothesis space is more expressive, it has a substantially larger representation. here we develop two bounds—both for linear classifiers on a restricted problem. The first upper bounds the generalization error for learning locally by assuming various degrees of separability. The second provides an improved generalization bound for globally learned classifiers by assuming separability in the more expressive global hypothesis space.

We begin by defining the *growth function* to measure the effective size of the hypothesis space.

Definition 6.1 (Growth Function) *For a given hypothesis class $\mathcal{H}$ consisting of functions $h : \mathcal{X} \to \mathcal{Y}$, the* growth function, *$\mathcal{N}_{\mathcal{H}}(m)$, counts the maximum number of ways to label any data set of size m:*

$$\mathcal{N}_{\mathcal{H}}(m) = \sup_{\mathbf{x}_1,\ldots,\mathbf{x}_m \in \mathcal{X}^m} |\{(h(\mathbf{x}_1),\ldots,h(\mathbf{x}_m))\,|h \in \mathcal{H}\}|$$

The well-known VC-style generalization bound expresses expected error, ϵ, of the *best* hypothesis, h_{opt} on unseen data. In the following theorem adapted from [Anthony and Bartlett, 1999][Theorem 4.2], we directly write the growth function into the bound,

Theorem 6.2 *Suppose that $\mathcal{H}$ is a set of functions from a set $\mathcal{X}$ to a set $\mathcal{Y}$ with growth function $\mathcal{N}_{\mathcal{H}}(m)$. Let $h_{\text{opt}} \in \mathcal{H}$ be the hypothesis that minimizes sample error on a sample of size m drawn from an unknown, but fixed probability distribution. Then, with probability $1-\delta$*

$$\epsilon \leq \epsilon_{\text{opt}} + \sqrt{\frac{32(\log(\mathcal{N}_{\mathcal{H}}(2m)) + \log(4/\delta))}{m}}. \qquad (2)$$

For simplicity, we first describe the setting in which a separate function is learned for each of a fixed number, c, of output variables (as in Section 5.1). Here, each example has c components in input $\mathbf{x} = (\mathbf{x}_1,\ldots,\mathbf{x}_c) \in \mathbb{R}^d \times \ldots \times \mathbb{R}^d$ and output $\mathbf{y} = (y_1,\ldots,y_c) \in \{0,1\}^c$.

Given a dataset $\mathbf{D}$, the aim is to learn a set of linear scoring functions $f_i(\mathbf{x}_i) = \mathbf{w}_i\mathbf{x}_i$, where $\mathbf{w}_i \in \mathbb{R}^d$ for each $i = 1,\ldots,c$. For LO and L+I, the setting is simple: find a set of weight vectors that, for each component, satisfy $y_i\mathbf{w}_i\mathbf{x}_i > 0$ for all examples $(\mathbf{x},\mathbf{y}) \in \mathbf{D}$. For IBT, we find a set of classifiers such that $\sum_i y_i\mathbf{w}_i\mathbf{x}_i > \sum_i y'_i\mathbf{w}_i\mathbf{x}_i$ for all $y' \neq y$ (and that satisfy the constraints, $y' \in \mathcal{C}(\mathcal{Y}^c)$).

As previously noted, when learning local classifiers independently (LO and L+I), one can only guarantee convergence when each local problem is separable – however, it is often the case that global constraints render these problems inseparable. Therefore, there is a lower bound, ϵ_{opt}, on the optimal error achievable. Since each component is a separate learning problem, the generalization error is thus

Corollary 6.3 *When $\mathcal{H}$ is the set of separating hyperplanes in $\mathbb{R}^d$,*

$$\epsilon \leq \epsilon_{\text{opt}} + \sqrt{\frac{32(d\log((em/d)) + \log(4/\delta))}{m}}. \qquad (3)$$

Proof sketch: We show that $\mathcal{N}_{\mathcal{H}}(m) \leq (em/d)^d$ when $\mathcal{H}$ is the class of threshold linear functions in d dimensions. $\mathcal{N}_{\mathcal{H}}(m)$ is precisely the maximum number of continuous regions an arrangement of m halfspaces in $\mathbb{R}^d$, which is $2\sum_{i=1}^{d}\binom{m-1}{i} \leq 2(e(m-1)/d)^d$. For $m > 1$, the result holds. See [Anthony and Bartlett, 1999][Theorem 3.1] for details. ■

On the other hand, when learning collectively with IBT, examples consist of the full vector $\mathbf{x} \in \mathbb{R}^{cd}$. In this setting, convergence is guaranteed (if, of course, such a function

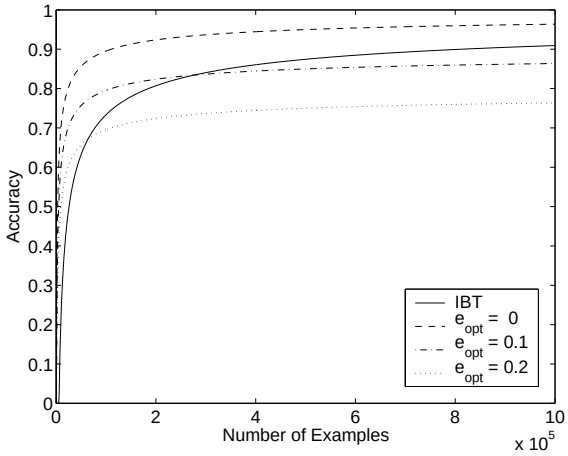


Figure 5: The VC-style generalization bounds predict that IBT will eventually outperform LO if the local classifiers are unable to find consistent classification ($\epsilon_{\text{opt}} > 0.0$, accuracy < 1). However, if the local classifiers are learnable ($\epsilon_{\text{opt}} = 0.0$, accuracy $= 1$), LO will perform well.

exists). Thus, the optimal error when training with IBT is $\epsilon_{\text{opt}} = 0$. However, the output of the global classification is now the entire output vector. Therefore, the growth function must account for exponentially many outputs.

Corollary 6.4 *When $\mathcal{H}$ is the set of decision functions over $\{0,1\}^c$, defined by $\text{argmax}_{\mathbf{y}' \in \mathcal{C}(\{0,1\}^c)} \sum_{i=1}^{c} y_i \mathbf{w}_i \mathbf{x}_i$, where $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_c) \in \mathbb{R}^{cd}$,*

$$\epsilon \leq \sqrt{\frac{32(cd\log(em/cd) + c^2 d + \log(4/\delta))}{m}}. \qquad (4)$$

Proof sketch: In this setting, we must count the effective hypothesis space – which is the effective number of different classifiers in weight space, $\mathbb{R}^{cd}$. As before, this is done by constructing an arrangement of halfspaces in the weight space. Specifically, each halfspace is defined by a single $((\mathbf{x}, \mathbf{y}), \mathbf{y}')$ pair that defines the region where $\sum_i y_i \mathbf{w}_i \mathbf{x}_i > \sum_i y'_i \mathbf{w}_i \mathbf{x}_i$. Because there are potentially $\mathcal{C}(2^c) \leq 2^c$ output labels and the weight space is cd-dimensional, the growth function is the size of the arrangement of $c2^c$ halfspaces in $\mathbb{R}^{cd}$. Therefore $\mathcal{N}_{\mathcal{H}}(m) \leq (em2^c/cd)^{cd}$. ■

Figure 5 shows a comparison between these two bounds, where the generalization bound curves on accuracy are shown for IBT (Corollary 6.4) and for LO and L+I (Corollary 6.3) with $\epsilon_{\text{opt}} \in \{0.0, 0.1, 0.2\}$. One can see that when separable, the accuracy=1 curve ($\epsilon_{\text{opt}} = 0.0$) in the figure outperforms IBT. However, when the problems are locally inseparable, IBT will eventually converge, whereas LO and L+I will not – these results match the synthetic experiment results in Figure 2. Notice the relationship between κ and ϵ_{opt}. When $\kappa = 0$, both the local and global problems are separable and $\epsilon_{\text{opt}} = 0$ As κ increases, the global problem remains separable and the local problems are inseparable ($\epsilon_{\text{opt}} > 0$).

7 Conclusion

We studied the tradeoffs between three common learning schemes for structured outputs, i.e. learning without the knowledge about structure (LO), using inference only after learning (L+I), and learning with inference feedback (IBT). We provided experiments on both real-world and synthetic data as well as a theoretical justification that support our main clams. – first, when the local classification is linearly separable, L+I outperforms IBT, and second, as the local problems become more difficult and are no longer linearly separable, IBT outperforms L+I, but only with sufficient number of training examples. In the future, we will seek a similar comparison for the more general setting where nontrivial interaction between local classifiers is allowed, and thus, local separability does not imply global separability.

8 Acknowledgments

We grateful Dash Optimization for the free academic use of Xpress-MP. This research is supported by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program, a DOI grant under the Reflex program, NSF grants ITR-IIS-0085836, ITR-IIS-0085980 and IIS-9984168, and an ONR MURI Award.

References

[Anthony and Bartlett, 1999] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[Carreras and Màrquez, 2003] X. Carreras and Lluís Màrquez. Online learning via global feedback for phrase recognition. In *Advances in Neural Information Processing Systems 15*, 2003.

[Collins, 2002] M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, 2002.

[Har-Peled *et al.*, 2003] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: A new approach to multiclass classification and ranking. In *Advances in Neural Information Processing Systems 15*, 2003.

[Kingsbury and Palmer, 2002] P. Kingsbury and M. Palmer. From Treebank to PropBank. In *Proceedings of LREC*, 2002.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML-01*, pages 282–289, 2001.

[Marcus *et al.*, 1993] M. P. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

[McCallum *et al.*, 2000] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of ICML-00*, Stanford, CA, 2000.

[Punyakanok and Roth, 2001] V. Punyakanok and D. Roth. The use of classifiers in sequential inference. In *Advances in Neural Information Processing Systems 13*, 2001.

[Punyakanok *et al.*, 2004] V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Semantic role labeling via integer linear programming inference. In *Proceedings of COLING-04*, 2004.

[Ramshaw and Marcus, 1995] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, 1995.

[Roth and Yih, 2004] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL-2004*, pages 1–8, 2004.

[Taskar *et al.*, 2004] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Advances in Neural Information Processing Systems 16*, 2004.