Learning and Inference over Constrained Output

Vasin Punyakanok Dan Roth Wen-tau Yih Dav Zimak

Presenter： Mingyang Liu（Applied math）

Three fundamentally different solutions to learn classifiers over structured output ★

- Local classifiers are learned and used to predict each output component separately (LO)
  - Learning: Find the hypothesis h: $\mathscr{X} \rightarrow \mathscr{Y}$ without constraints/structure in output. Cheaper computationally
  - Prediction: y= $argmax_y h(x)$
  - Searching space is small
  - Eg. SVM, perceptron, regression

Three fundamentally different solutions to learn classifiers over structured output ★

- Learning is decoupled from the task of maintaining structured output(L+I)
  - Learning step: Find the hypothesis h: $\mathscr{X} \rightarrow \mathscr{Y}$ without dependencies among y_i. Cheaper computationally.
  - Making decision step: predict the best structure y= $(y_1, \dots, y_T)$ with dependencies among y_i
  - Searching space is large(NP-hard)
  - Eg. Conditional models[McCallum *et al* 2000]
    - In the learning procedure, we learn single classifer $P(S_t = s_t | S_{t-1} = s_{t-1}, O_t = o_t)$, so there is not inference because there we do not build a classifer for the whole structure/sequence.
    - In the final decision step, put all the estimated parameters in the model and use them in Viterbi, which is a global inference algorithm, to predict the best sequence of states. The structure of the sequence is in this step. So L+I
  - Incorporating global constraints sometimes is not available, not needed, or just too expensive

Three fundamentally different solutions to learn classifiers over structured output ★

- Incorporating dependencies among the variables into the learning process(IBT)
  - Learning: Find the hypothesis h: $\mathscr{X} \rightarrow \mathscr{Y}$ with dependencies among y_i. MakIng learning more difficult
  - Making decision step: predict the best structure y= $(y_1, \ldots, y_T)$ with dependencies among y_i
  - Searching space is large
  - Eg. CRF[Lafferty *et al.*, 2001]
    - $$\log p\left(\boldsymbol{w} \mid D; \sigma^2\right) = -\frac{1}{\sigma^2}||\boldsymbol{w}||^2 + \sum_{n=1}^{N}\left[\boldsymbol{w}^\top \Phi(x_n, y_n) - \log \sum_{y' \in \mathcal{Y}} \exp\left[\boldsymbol{w}^\top \Phi(x_n, y')\right]\right]$$

Lots of choice , constraints

L+I v.s IBT in Chunking

- Goal: identification of parts of speech
- Given $o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}$
- Classifer 1(start of chunk): [[[[[
- Classifer 2(end of chunk):]]]]]
- Inference(constraints): [ ] [ ]

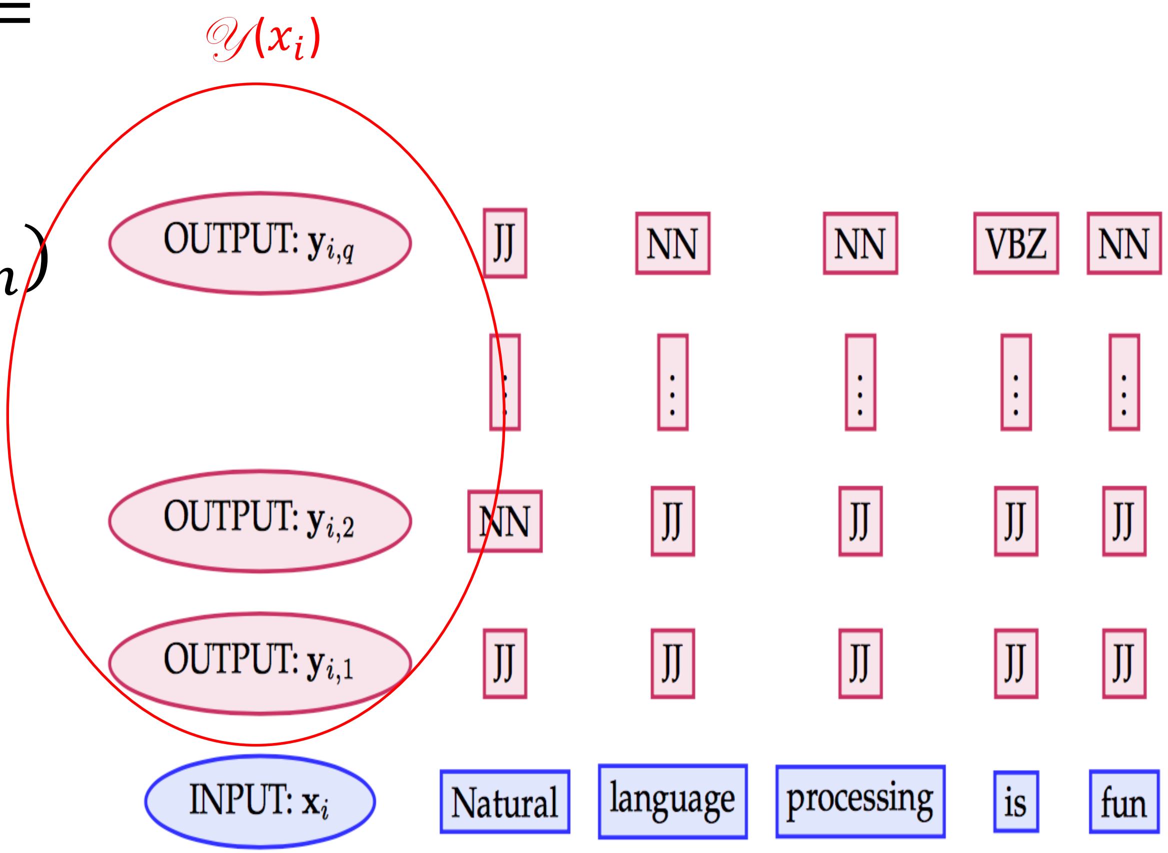
learning independent classifiers(LO, L+I)

vs ?

Inference based training(IBT)

Definition: Structured classification problem

- Given an observation of input variable $\boldsymbol{X} = (X_1, \ldots, X_n)$=$(x_1, \ldots, x_n)$
- Find 'best' assignment y for $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$
- y is consistent with structure on $\boldsymbol{Y}$
- This structure can be thought of as constraining the output space $\mathscr{Y}^n$ to a smaller space C($\mathscr{Y}^n$) $\subseteq$ $\mathscr{Y}^n$



Definition: Structure output classifier ★

- Local scoring functions $f_y(\boldsymbol{x}, t)$, f_y: $\mathscr{X}^{\mathrm{n}} \times \{1, \dots, \mathrm{n}\} \rightarrow \mathrm{R}$
  - Represent the score for $Y_t = y \in \mathscr{Y}$

- Global scoring function f: $\mathscr{X}^{\mathrm{n}} \times \mathscr{Y}^n \rightarrow \mathrm{R}$
  - $f(\boldsymbol{x}, \boldsymbol{y}) = f\big(\boldsymbol{x}, (y_1, \dots, y_n)\big) = \sum_{t=1}^{n} f_{y_t}(\boldsymbol{x}, t)$
  - Eg. Dependency Parsing
    - Find the highest scoring dependency tree, from the space of all dependency trees of N words.
    - Learn a model to score edge (i,j) of a candidate tree $s(i, j) = w \cdot f(i, j)$
    - Score of a dependency tree is sum of score of its edges $s(x, y) = \sum_{(i,j) \in y} s(i, j) = \sum_{(i,j) \in y} w \cdot f(i, j)$

- Structured output classifier h: $\mathscr{X}^{\mathrm{n}} \rightarrow \mathscr{Y}^n$
  - $h(\boldsymbol{x}) = argmax_{y' \in C(\mathscr{Y}^n)} f(\boldsymbol{x}, y')$

Definition: Linear representation

- Linear local scoring function $f_y(\boldsymbol{x}, t) = \alpha^y \cdot \Phi^y(\boldsymbol{x}, t)$
  - α^y is weight vector , $\Phi^y(\boldsymbol{x}, t)$ is feature vector

- Linear global scoring function $f(\boldsymbol{x}, \boldsymbol{y}) = \alpha \cdot \Phi(\boldsymbol{x}, \boldsymbol{y})$
  - $\alpha, \Phi(x, y) \in R^{|\mathscr{Y}|}$
  - $\Phi(\boldsymbol{x}, \boldsymbol{y})$=($\Phi^1(\boldsymbol{x}, \boldsymbol{y})$,…, $\Phi^{|\mathscr{Y}|}(\boldsymbol{x}, \boldsymbol{y})$)
  - $\Phi^y(\boldsymbol{x}, \boldsymbol{y}) = \sum_{t=1}^{n} \Phi^{y_t}(\boldsymbol{x}, t) I_{\{y_t = y\}}$ for class y

- Structured output classifier h: $\mathscr{X}^{\mathrm{n}} \to \mathscr{Y}^n$
  - $h(\boldsymbol{x}) = argmax_{y' \in C(\mathscr{Y}^n)}\, \alpha \cdot \Phi(\boldsymbol{x}, \boldsymbol{y}')$

Online perceptron-style algorithm

No global constraints

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 $t = 1, \ldots, n_{\mathbf{y}}$ **do**

      $\hat{y}_t = \boxed{\text{argmax}_y}\, \boldsymbol{\alpha}^y \cdot \Phi^y(\mathbf{x}, t)$

      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**

    $\hat{\mathbf{y}} = \boxed{\text{argmax}_{\mathbf{y} \in \mathcal{C}(\mathcal{Y}^{n_{\mathbf{y}}})}}\, \boldsymbol{\alpha} \cdot \Phi(\mathbf{x}, \mathbf{y})$

    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

Having global constraints

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

Conjectures

- When local classification problems are easy: LO>L+I>IBT
  - Information from Structure is not necessary
- When local classification problems are getting harder: L+I>LO>IBT
  - Structure becomes more important
  - We also have decent classifiers learned locally
- When local classification problems are extremely harder: IBT>L+I>LO
  - It is unlikely that structure based inference can fix poor classifiers learned locally

Definition: Separability and Learnability

- A classifier, $f \in H$, globally separates a dataset D iff for all examples $(\boldsymbol{x}, \boldsymbol{y}) \in D, f(\boldsymbol{x}, \boldsymbol{y}) > f(\boldsymbol{x}, \boldsymbol{y}')$ for all $\boldsymbol{y}' \in \mathcal{Y}^n \backslash \boldsymbol{y}$
  - **All-vs-all**
- A classifier, $f \in H$, locally separates a dataset D iff for all examples $(\boldsymbol{x}, \boldsymbol{y}) \in D, f_{y_t}(\boldsymbol{x}, t) > f_y(\boldsymbol{x}, t)$ for all y$\in \mathcal{Y} \backslash y_t$ and for all t
  - 1-vs-all
- Learning algorithm $\mathscr{A} : D \rightarrow H$
- D is globally/locally learnable by $\mathscr{A}$ if there exists an $f \in H$ such that f globally/locally separates D

Relationships between local and global learning

- local separability implies global separability, but the inverse is not true
  - $f(\boldsymbol{x}, \boldsymbol{y}) = \sum_{t=1}^{n} f_{y_t}(\boldsymbol{x}, t)$>$\sum_{t=1}^{n} f_{y_t{'}}(\boldsymbol{x}, t)$= $f(\boldsymbol{x}, \boldsymbol{y}')$ for at least one t, $y_t' \neq y_t$

- local separability implies local and global learnability

- global separability implies global learnability, but not local learnability

Claim

- If the local classification tasks are separable, then L+I outperforms IBT

- If the task is globally separable, but not locally separable then IBT outperforms L+I only with sufficient examples.

Experiments

(Synthetic Data)

- Each example **x** = $(x_1, x_2, \ldots, x_c) \in R^d \times \ldots \times R^d$

- Binary label **y** = $(y_1, \ldots, y_c) \in \{0,1\}^c$ from
  - $\boldsymbol{y} = h(\boldsymbol{x}) = argmax_{\boldsymbol{y} \in C(\mathscr{Y})} \sum_i y_i f_i(x_i) - (1 - y_i) f_i(x_i)$

- $C(\mathscr{Y})$ is a random constraint on **y**

- Each f_i corresponds to a local classifier $y_i = g_i(x_i) = I_{f\,i(x_i) > 0}$

- The dataset generated from this hypothesis is globally linearly separable
  - Let f(**x**, **y***)= $\sum_i y_i f_i(x_i) - (1 - y_i) f_i(x_i)$. f(**x**, **y***)> f(**x**, **y'**) for all $\boldsymbol{y}' \in C(\mathscr{Y})\backslash$**y***from argmax.

Experiments

(vary the difficulty of local classification)

- Let fraction κ of the data where $h(\boldsymbol{x}) \neq g(\boldsymbol{x}) = (g_1(x_1), \dots, g_c(x_c))$
  - i.e. $g(x) \notin C(\mathscr{Y})$ because of constraint space

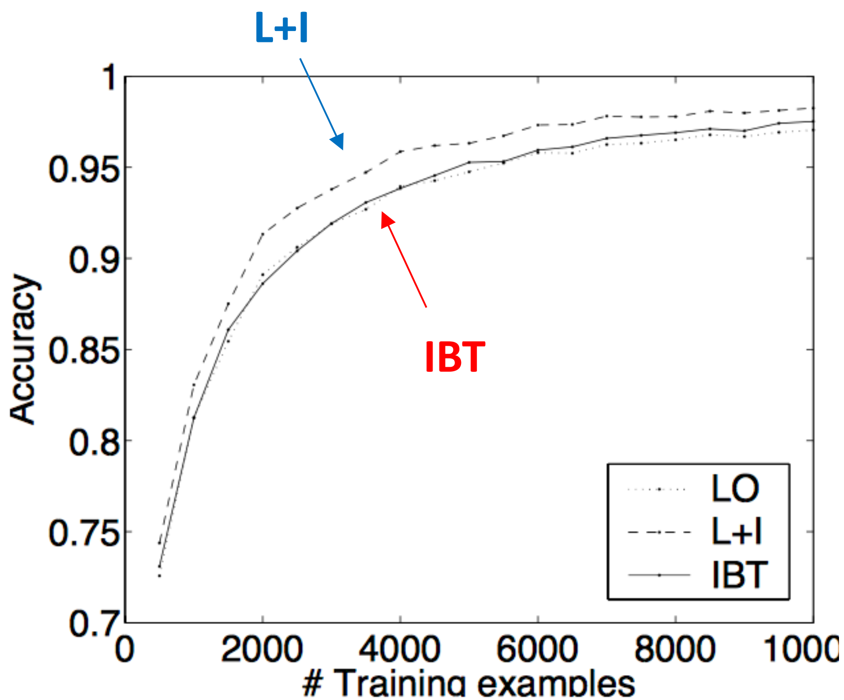
- We can regard κ as how many bracket appear in the single classifier but not exist after inference.

Given	$o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}$
Classifer 1(start of chunk):	[[[[[
Classifer 2(end of chunk):	]]]]]
Inference(constraints):	[ ] [ ]

Black brackets are

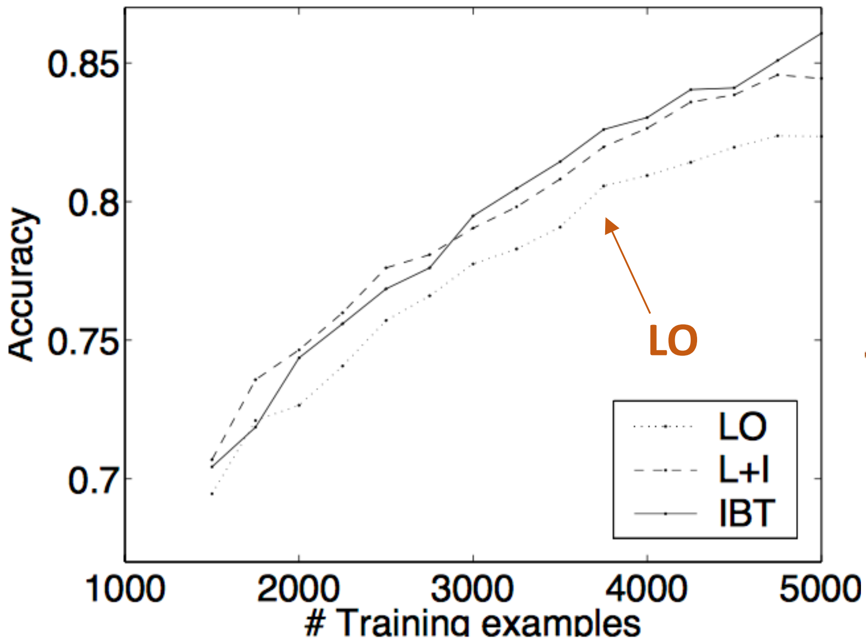
- chosen by local classifiers
- rejected by constraints
- Indicating quality of local classifers

Performance



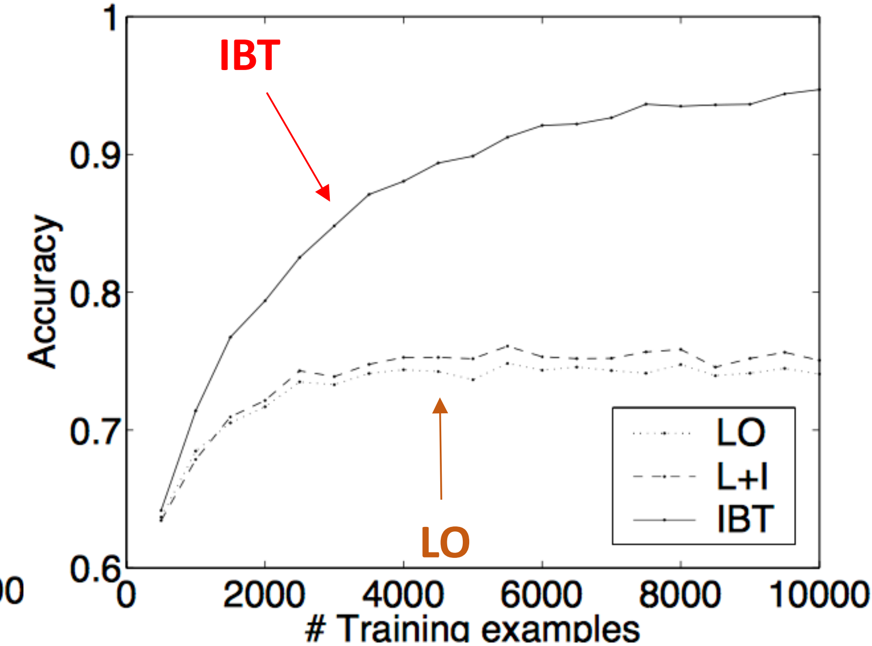
(a) $\kappa = 0, d = 100$

locally linearly separable



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

not totally locally linearly separable



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

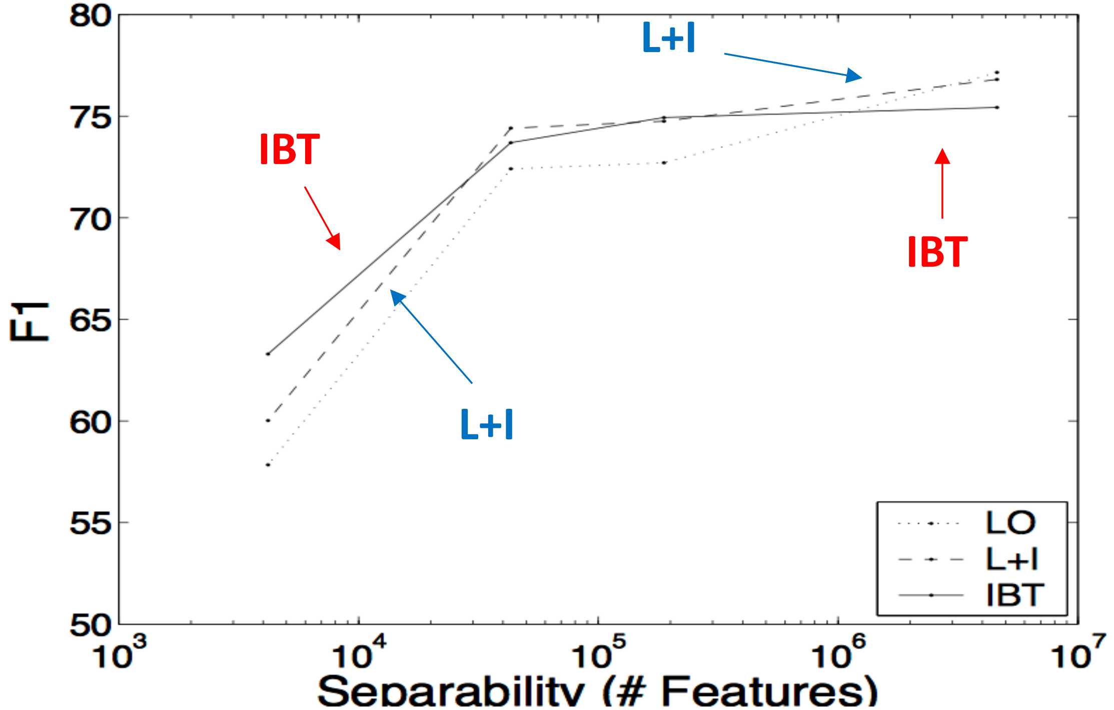
most difficult local classification tasks

In all cases, inference helps

Experiments

(Real-World Data)

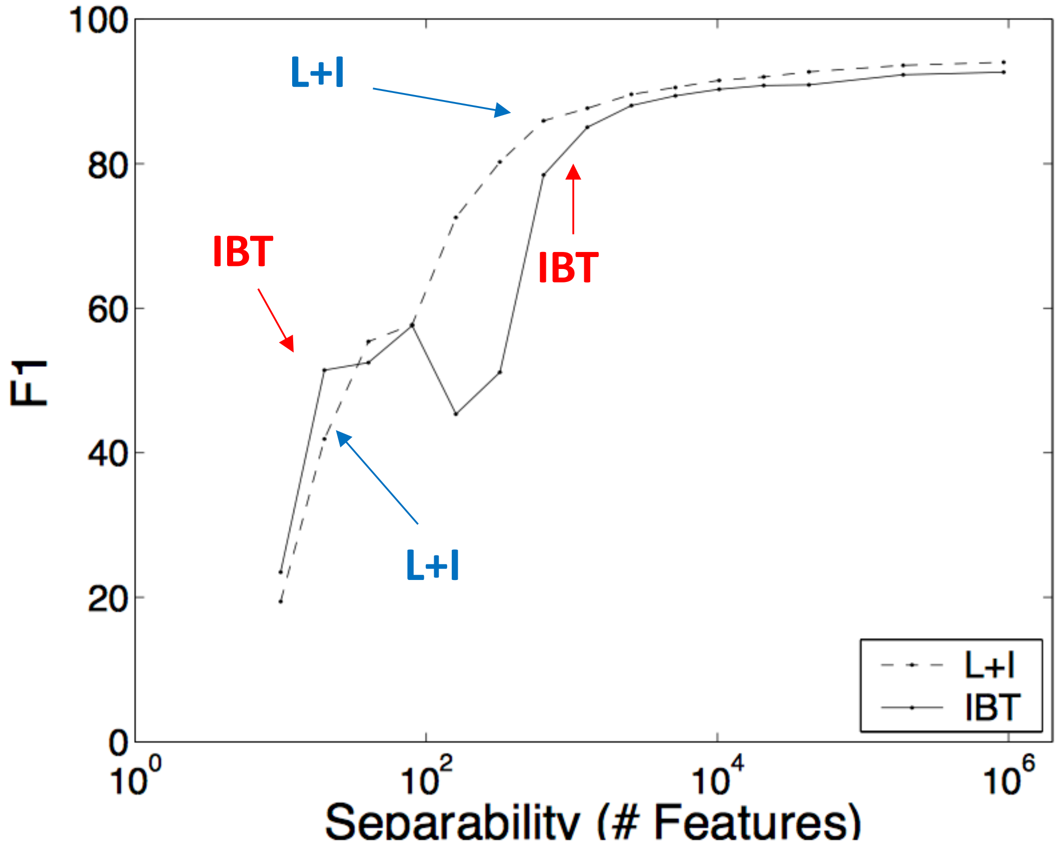
- Semantic-Role Labeling
  - To identify, for each verb in the sentence, all the constituents which fill a semantic role, and determine their argument types.
  - Structural constraints are necessary to ensure, for example, that no arguments can overlap or embed each other.



More features
⇒more separable
⇒local classifiers are easy to learn

Experiments
(Real-World Data)

- Noun Phrase Labeling
  - identification of phrases or of words that participate in a syntactic relation-ship



Similarly, only when the problem becomes difficult IBT > L+I

Bound Prediction

- 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.

- While the global hypothesis space is more expressive, it has a substantially larger representation.(Need more data)

Well-known VC-style generalization bound

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:*

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

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 $1 - \delta$*

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

Upper bounds of generalization error for learning locally

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

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

Improved generalization bound for globally learned classifiers

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

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