Self-Paced Learning for Latent Variable Models

Kumar M P, Packer B, Koller D. //NIPS 2010

Y.Bengio et al. Curriculum learning In ICML 2009

★ Learning from easier aspects of the task, and gradually increase the difficulty level

★ Expected two advantages:
- Help find a better local minima (as a regularizer)
- Speed the convergence of training towards the global minimum (for convex problem)

★ Basic steps:
- Sort samples according to certain “easiness” measure
- Gradually add samples into training from easy to complex

Latent SSVM

$\Phi(x , y , h)$

Latent SSVM

For any given w, the value of ξi can be shown to be an upper bound on the risk Δ(yi,y ˆi(w))

$$\min_{\mathbf{w},\xi_i \geq 0} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i,$$

$$\text{s.t.} \quad \max_{h_i \in \mathcal{H}} \mathbf{w}^\top \left(\Phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i) - \Phi(\mathbf{x}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{h}}_i)\right) \geq \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) - \xi_i,$$

$$\forall \hat{\mathbf{y}}_i \in \mathcal{Y}, \forall \hat{h}_i \in \mathcal{H}, i = 1, \cdots, n.$$

Algorithm 2 *The* CCCP *algorithm for parameter estimation of latent* SSVM.

input $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$, $\mathbf{w}_0$, ϵ.

1: $t \leftarrow 0$

2: **repeat**

3: Update $\mathbf{h}_i^* = \text{argmax}_{h_i \in \mathcal{H}} \mathbf{w}_t^\top \Phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i)$.

4: Update $\mathbf{w}_{t+1}$ by fixing the hidden variables for output $\mathbf{y}_i$ to $\mathbf{h}_i^*$ and solving the corresponding SSVM problem. Specifically,

$\mathbf{w}_{t+1} = \text{argmin}_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_i \max\{0, \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) + \mathbf{w}^\top(\Phi(\mathbf{x}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{h}}_i) - \Phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i^*))\}$.

5: $t \leftarrow t + 1$.

6: **until** Objective function cannot be decreased below tolerance ϵ.

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \left(r(\mathbf{w}) + \sum_{i=1}^{n} f(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) \right)$$

f(.) is the negative log-likelihood for EM or an upper bound on the risk for latent SSVM (or any other criteria for parameter learning).

binary variables vi that indicate whether the ith sample is easy or not.
Only easy samples contribute to the objective function.

$$(\mathbf{w}_{t+1}, \mathbf{v}_{t+1}) = \underset{\mathbf{w}\in\mathbb{R}^d, \mathbf{v}\in\{0,1\}^n}{\operatorname{argmin}} \left(r(\mathbf{w}) + \sum_{i=1}^{n} v_i f(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) - \frac{1}{K} \sum_{i=1}^{n} v_i \right)$$

$$(\mathbf{w}_{t+1}, \mathbf{v}_{t+1}) = \underset{\mathbf{w}\in\mathbb{R}^d, \mathbf{v}\in\{0,1\}^n}{\operatorname{argmin}} \left(r(\mathbf{w}) + \sum_{i=1}^{n} v_i f(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) - \frac{1}{K} \sum_{i=1}^{n} v_i \right)$$

Algorithm 3 *The self-paced learning algorithm for parameter estimation of latent* SSVM.

input $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$, $\mathbf{w}_0$, K_0, ϵ.

1: $t \leftarrow 0$, $K \leftarrow K_0$.
2: **repeat**
3: Update $h_i^* = \operatorname{argmax}_{h_i \in \mathcal{H}} \mathbf{w}_t^\top \Phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i)$.
4: Update $\mathbf{w}_{t+1}$ by using ACS to minimize the objective $\frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n} \sum_{i=1}^{n} v_i \xi_i - \frac{1}{K} \sum_{i=1}^{n} v_i$ subject to the constraints of problem (2) as well as $\mathbf{v} \in \{0, 1\}^n$.
5: $t \leftarrow t + 1$, $K \leftarrow K/\mu$.
6: **until** $v_i = 1, \forall i$ and the objective function cannot be decreased below tolerance ϵ.

实验结果

Experiments

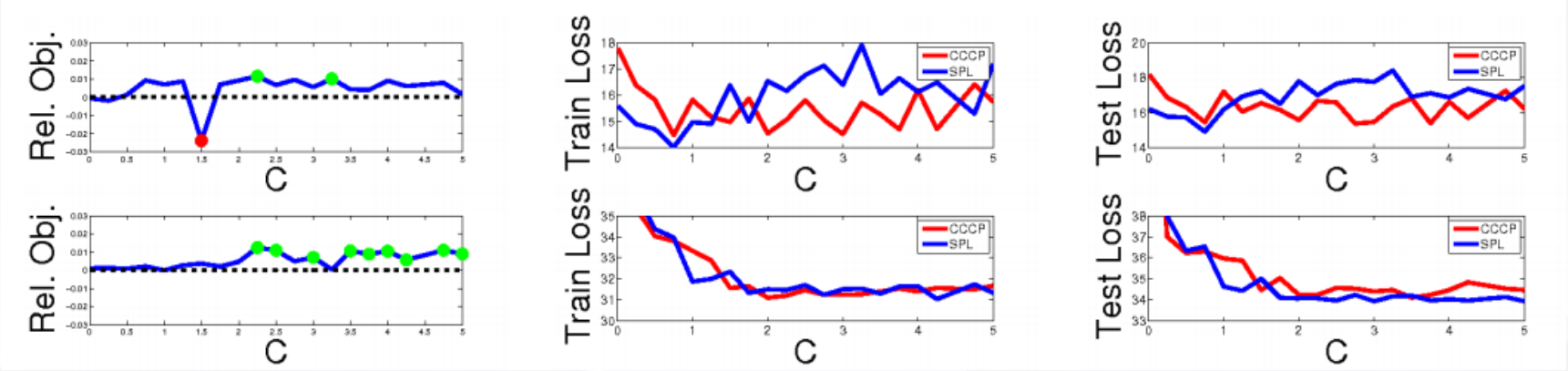
- Noun Phrase Coreference
- Motif Finding

Noun Phrase Coreference

Given the occurrence of all the nouns in a document, the goal of noun phrase coreference is to provide a clustering of the nouns such that each cluster refers to a single object.

MITRE score

Pair wise score



$$(obj_{cccp} - obj_{spl})/obj_{cccp}$$

Motif Finding

binary classification of DNA sequences

For all experiments we use C = 150 and ε = 0.001 (the large size of the dataset made cross-validation highly time consuming)

(a) Objective function value

CCCP	92.77 ± 0.99	$\mathbf{106.50 \pm 0.38}$	94.00 ± 0.53	116.63 ± 18.78	75.51 ± 1.97
SPL	$\mathbf{92.37 \pm 0.65}$	106.60 ± 0.30	$\mathbf{93.51 \pm 0.29}$	$\mathbf{107.18 \pm 1.48}$	$\mathbf{74.23 \pm 0.59}$

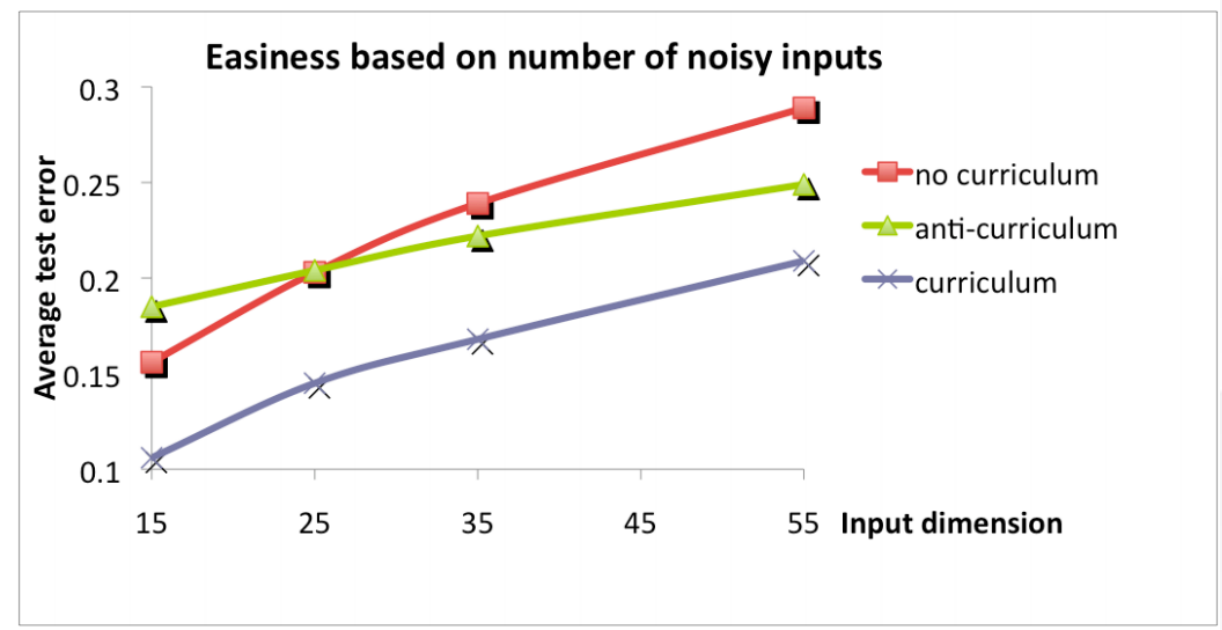
(b) Training error (%)

CCCP	27.10 ± 0.44	$\mathbf{32.03 \pm 0.31}$	26.90 ± 0.28	34.89 ± 8.53	20.09 ± 0.81
SPL	$\mathbf{26.94 \pm 0.26}$	32.04 ± 0.23	$\mathbf{26.81 \pm 0.19}$	$\mathbf{30.31 \pm 1.14}$	$\mathbf{19.52 \pm 0.34}$

(c) Test error (%)

CCCP	27.10 ± 0.36	$\mathbf{32.15 \pm 0.31}$	27.10 ± 0.37	35.42 ± 8.19	20.25 ± 0.65
SPL	$\mathbf{27.08 \pm 0.38}$	32.24 ± 0.25	$\mathbf{27.03 \pm 0.13}$	$\mathbf{30.84 \pm 1.38}$	$\mathbf{19.65 \pm 0.39}$

one could argue that difficult examples can be more informative than easy examples. Here the difficult examples are probably not useful because they confuse the learner rather than help it establish the right location of the decision surface.



THANK YOU