Multi-core Structural SVM Training

Kai-Wei Chang, Vivek Srikumar and Dan Roth

Motivation

- Many applications require structured decisions.
- Global decisions where local decisions play a role but there are mutual dependencies on their outcome.
- It is essential to make coherent decisions in a way that takes the interdependencies into account.

Part-of-Speech tagging (sequential labeling)

- Input: a sequence of words $\{x_1, x_2, \dots x_n\}$.
- Output: POS tags $\{y_1, y_2, \dots y_n\}$, $y_i \in \{NN, VBZ, \dots\}$ "A cat chases a mouse" => "DT NN VBZ DT NN".
- Assignment to y_i can depend on both x_i and y_{i-1}.
- Feature vector $\phi(x,y)$ defined on both input and output variables: e.g., "x_i: Cat", "y_{i-1}: VBZ.

Structured Prediction Model

- Structured prediction: predicting a structured output variable y based on the input variable x.
- $y = \{y_1, y_1, \dots y_n\}$ variables form a structure: sequences, clusters, trees, or arbitrary graphs.
- Various approaches have been proposed to learn structured prediction models [Joachims et. al. 09, Chang and Yih 13, Lacoste-Julien et. al. 13] but they are single-threaded.
- DEMI-DCD: a multi-threaded algorithm for training structural SVM.

Advantages:

- Requires little synchronization between threads
  ⇒ fully utilizes the power of multiple cores.
- Makes multiple updates on the structures discovered by the loss-augmented inference
  ⇒ fully utilizing the available information

Inference

$$\underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\ \mathbf{w}\cdot\phi(\mathbf{x},\mathbf{y})$$

Weight parameters

Set of allowed structures often specified by constraints

Features on input-output

- Efficient inference algorithms have been proposed for some specific structures.
- Integer linear programing (ILP) solver can deal with general structures.

Abstract

Many problems can be framed as structured prediction problems. Structural support vector machines (SVM) is a popular approach for training structured predictors. In structural SVM, learning is done by alternating between an inference (prediction) phase and a model update phase. The inference phase selects candidate structures for all training examples and then the model is updated based on these structures. This paper develops an efficient multi-core implementation for structural SVM. We extend the dual coordinate descent approach by decoupling the model update and inference phases into different threads. We prove that our algorithm not only converges but also fully utilizes all available processors to speed up learning.

Structural SVM

$$\min_{w,\xi}\ \frac{1}{2}w^Tw + C\sum_i \ell(\xi_i)$$

$$s.t.\ w^T\Phi(x_i,y_i) - w^T\Phi(x_i,y) \geq \Delta(y_i,y) - \xi_i \quad \forall i, y \in \mathcal{Y}_i.$$

Score of gold structure | Score of predicted structure | Loss function | Slack variable

- We use $\ell(\xi) = \xi^2$ and solve the dual problem:

$$\min_{\alpha \geq 0} D(\alpha), \text{ and}$$

$$D(\alpha) \equiv \frac{1}{2}\left\|\sum_{\alpha_{i,y}} \alpha_{i,y}\phi(y,y_i,x_i)\right\|^2 + \frac{1}{4C}\sum_i\left(\sum_y \alpha_{i,y}\right)^2 - \sum_{i,y}\Delta(y,y_i)\alpha_{i,y},$$

where $\phi(y,y_i,x_i) = \phi(y_i,x_i) - \phi(y,x_i)$.

- # α variables can be exponentially large.
- Relationship between w^* and α^*

$$w^* = \sum_{i,y}\alpha^*_{i,y}\phi(y,y_i,x_i).$$

For linear model: we maintain the relationship between w and α [Hsieh et.al. 08].

Active Set Selection

- Maintain an active set A: Identify $\alpha_{i,y}$ that will be non-zero in the end of optimization process.
- In a single-thread implementation, training consists of two phases:

1. Select and maintain A (active set selection step).
   - Require solving a loss-augmented inference problem for each x_i

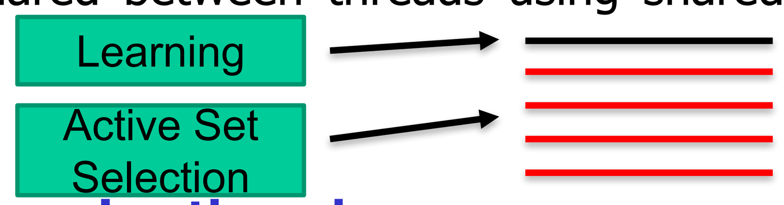
   $$\max_{y\in\mathcal{Y}_i}\ w^T\phi(x_i,y) + \Delta(y_i,y)$$

   - Solving loss-augmented inferences is usually the bottleneck.
2. Update the values of $\alpha_{(i,y)} \in A$ (learning step).

Demi-DCD

Decouple **Model-update** and **Inference** with **Dual Coordinate Descent.**

- Split training data into $B_1, B_2, \dots B_{p-1}$. (p:#threads).
- Active set selection (Inference) thread j: select and maintain the active set A_i for each example i in B_j.
- Learning thread: loop over all examples and update model w.
- A and w are shared between threads using shared memory buffers.



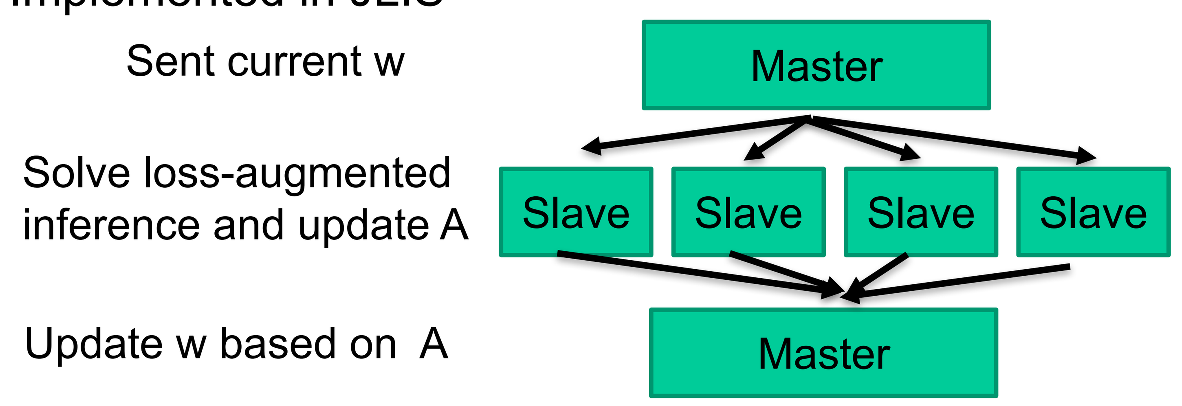
More about the learning thread

- Sequentially visit x_i and update $\alpha_{i,y} \in A_i$.
- Solve a sub-problem to update $\alpha_{i,y}$ and w
- Shrinking heuristic: remove $\alpha_{i,y}$ based on gradient.

Other Multi-core Approaches

A Master-Slave architecture (MS-DCD):

Implemented in JLIS



Parallel Structured Perceptron (SP-IPM)

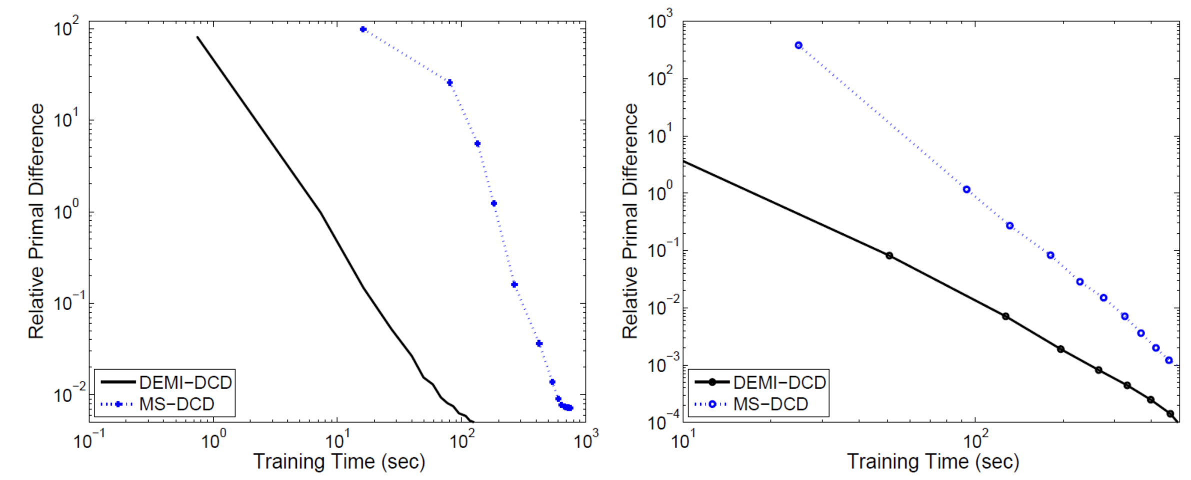
[McDonald et al 10]:

1. Split data into p parts.
2. Train Structured Perceptron on data blocks in parallel.
3. Mixed the models and use the mixed model as the initialization in Step 2.

Experiment Settings

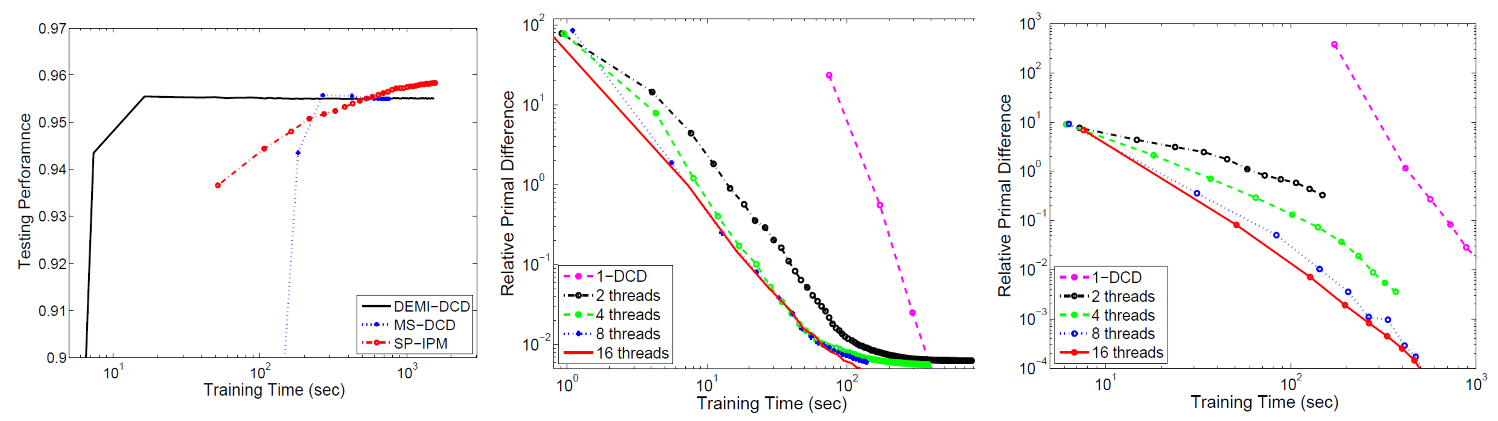
- POS tagging (POS-WSJ):
  - Assign POS label to each word in a sentence.
  - We use standard Penn Treebank Wall Street Journal corpus with 39,832 sentences.
- Entity and Relation Recognition (Entity-Relation):
  - Assign entity types to mentions and identify relations among them.
  - 5,925 training samples.
  - Inference is solved by an ILP solver.

Convergence on Primal Function Value



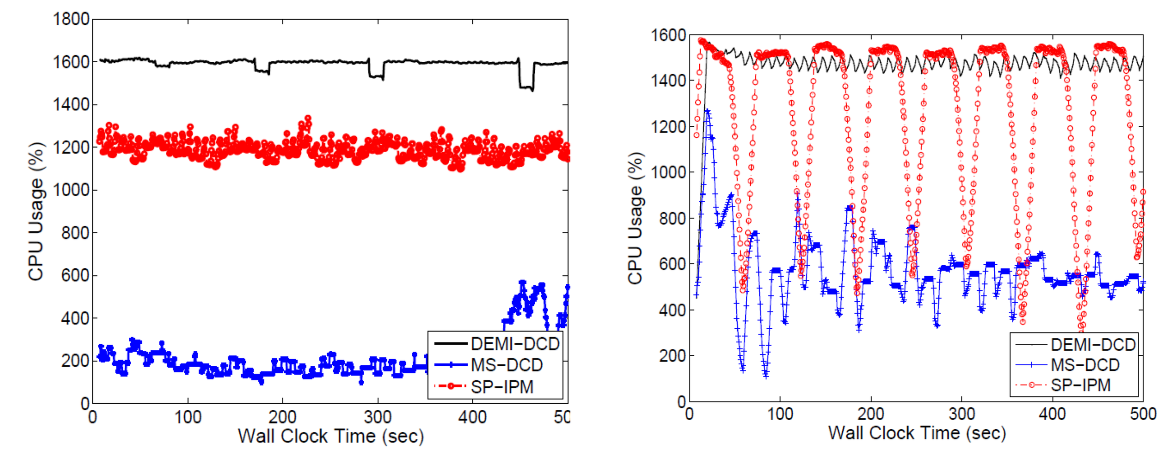
POS- WSJ   Entity-Relation

Test Performance



POS- WSJ   Entity   Relation

Moving average of CPU Usage



POS- WSJ   Entity-Relation

The code will be released at: **http://cogcomp.cs.illinois.edu/page/software**

This research is sponsored by DARPA and an ONR Award