Journal of Machine Learning Research 1 (2015) 1-?? Submitted 1/15; Published 1/15

Dissolve$^{\text{struct}}$– A Library for Distributed Structured Prediction

Tribhuvanesh Orekondy TOREKOND@STUDENT.ETHZ.CH
ETH Zurich

Martin Jaggi JAGGI@INF.ETHZ.CH
ETH Zurich

Aurelien Lucchi AURELIEN.LUCCHI@INF.ETHZ.CH
ETH Zurich

Editor: t.b.d.

Abstract

This paper describes DISSOLVE$^{\text{struct}}$, a modular and flexible open source software package for distributed training of structured prediction models, such as structured SVMs. Project website: github.com/dalab/dissolve.

We support a broad range of applications, and interfaces to scala, java and python. Our framework is empowered by the fault tolerant SPARK computing platform, and automatically adopts to the existing tradeoffs of computation vs communication cost on real world systems. The proposed distributed algorithm combines the recent communication efficient CoCoA scheme (Jaggi et al., 2014) with the state of the art primal-dual structured prediction solvers (Lacoste-Julien et al., 2013), and improves further by adding some new ideas for caching oracle answers. The framework allows approximate inference, and provides a similar standard interface as SVM$^{\text{struct}}$ for the user.

Keywords: Structured Prediction, Structured SVM, Distributed Training

1. Introduction

Structured prediction has gained a lot of popularity over the past few years due to its ability to process structured objects such as images or text documents.

The structured support vector machine (SSVM) is a particularly successful variant of this approach that can be optimized in various ways, including a cutting-plane, stochastic gradient descent or a primal-dual scheme such as MARTIN's paper. The latter is especially appropriate for large-scale problems ...
Remainder omitted in this sample. See http://www.jmlr.org/papers/ for full paper.

2. Structured SVM formulation

A structured model predicts the labeling Y for a given input X by maximizing some score function $S_{\mathbf{w}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, i.e.,

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\arg\max}\, S_{\mathbf{w}}(Y) = \underset{Y \in \mathcal{Y}}{\arg\max}\, \mathbf{w}^T \Psi(X, Y), \tag{1}$$

where $\mathbf{w}$ are the model parameters and $\Psi(X, Y)$ is the *feature map* corresponding to the input X and the labeling Y.

Given a set of N training examples $\mathcal{D} = ((X^1, Y^1), \ldots, (X^N, Y^N))$ where $X^i \in \mathcal{X}$ and $Y^i \in \mathcal{Y}$ is the associated labeling, the learning task consists in finding model parameters $\mathbf{w}$ that achieve low empirical loss subject to some regularization. This is usually formulated as a quadratic program (QP) with soft margin constraints **?**:

$$\begin{aligned} &\min_{\mathbf{w},\boldsymbol{\xi}\geq\mathbf{0}} \frac{\lambda}{2} \left|\left|\mathbf{w}\right|\right|^2 + \sum_{n=1}^{N} \xi_n \\ &\text{s.t. } \forall\, n : S_{\mathbf{w}}(Y^n) \geq \max_{Y \in \mathcal{Y}_n} (S_{\mathbf{w}}(Y) + \Delta(Y^n, Y)) - \xi_n, \end{aligned} \tag{2}$$

where $\mathcal{Y}_n$ is the set of all possible labelings for example n, the constant λ controls the trade-off between margin and training error, and the *task loss* Δ measures the closeness of any inferred labeling Y to the ground truth labeling Y^n.

TALK ABOUT DBCFW: description + pseudo-code

3. Software package

The use of the DISSOLVE$^{\text{struct}}$ requires users to implement the following functions:

- The joint feature map $\Psi(X, Y)$ which encodes the input/output pairs.
- The structured loss function $\Delta(Y_i, Y)$.
- A maximization oracle which computes the most violating constraint by solving Eq. X.
- A prediction function that computes X. This operation is usually performed with the maximization oracle.

Acknowledgments

...

Thank for all the students...

Appendix A.

In this appendix we prove the following theorem from Section 6.2:
Theorem *Let u, v, w be discrete variables such that v, w do not co-occur with u (i.e., $u \neq 0 \Rightarrow v = w = 0$ in a given dataset $\mathcal{D}$). Let N_{v0}, N_{w0} be the number of data points for which $v = 0, w = 0$ respectively, and let I_{uv}, I_{uw} be the respective empirical mutual information values based on the sample $\mathcal{D}$. Then*

$$N_{v0} \; > \; N_{w0} \;\; \Rightarrow \;\; I_{uv} \; \leq \; I_{uw}$$

with equality only if u is identically 0. ■
Proof. We use the notation:

$$P_v(i) \; = \; \frac{N_v^i}{N}, \quad i \neq 0; \quad P_{v0} \; \equiv \; P_v(0) \; = \; 1 - \sum_{i \neq 0} P_v(i).$$

These values represent the (empirical) probabilities of v taking value $i \neq 0$ and 0 respectively. Entropies will be denoted by H. We aim to show that $\frac{\partial I_{uv}}{\partial P_{v0}} < 0$....

Remainder omitted in this sample. See http://www.jmlr.org/papers/ for full paper.

References

Martin Jaggi, Virginia Smith, Martin Takáč, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-Efficient Distributed Dual Coordinate Ascent. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*, pages 3068–3076, 2014.

Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *ICML 2013 - Proceedings of the 30th International Conference on Machine Learning*, 2013.