Optimizing Multivariate Performance Measures for Learning Relation Extraction Models

Gholamreza Haffari
[2]Faculty of IT, Monash University
gholamreza.haffari@monash.edu

Ajay Nagesh[1,2,3]
[1]IITB-Monash Research Academy
ajaynagesh@cse.iitb.ac.in

Ganesh Ramakrishnan
[3]Dept. of CSE, IIT Bombay
ganesh@cse.iitb.ac.in

Abstract

We describe a novel max-margin learning approach to optimize non-linear performance measures for distantly-supervised relation extraction models. Our approach can be generally used to learn latent variable models under multivariate non-linear performance measures, such as F_β-score. Our approach interleaves Concave-Convex Procedure (CCCP) for populating latent variables with dual decomposition to factorize the original hard problem into smaller independent sub-problems. The experimental results demonstrate that our learning algorithm is more effective than the ones commonly used in the literature for distant supervision of information extraction models. On several data conditions, we show that our method outperforms the baseline and results in up to 8.5% improvement in the F_1-score.

1 Introduction

Rich models with latent variables are popular for many problems in natural language processing. In information extraction, for example, one needs to predict the relation labels $\mathbf{y}$ that an entity-pair $\mathbf{x}$ can have based on the hidden relation mentions $\mathbf{h}$, *i.e.*, the relation labels for occurrences of the entity-pair in a given corpus. However, these models are often trained by optimizing performance measures (such as conditional log-likelihood or error rate) that are not directly related to the task-specific non-linear performance measure, *e.g.*, the F_1-score. However, better models may be trained by optimizing the task-specific performance measure while allowing latent variables to adapt their values accordingly.

We present a large-margin method to learn parameters of latent variable models for a wide range of non-linear multivariate performance measures such as F_β. Our method can be applied to learning graphical models that incorporate inter-dependencies among the output variables either directly, or indirectly through hidden variables.

Large-margin methods have been shown to be a compelling approach to learn rich models detailing the inter-dependencies among the output variables, via optimizing loss functions decomposable over the *training instances* (Taskar et al., 2003; Tsochantaridis et al., 2004) or non-decompasable loss functions (Ranjbar et al., 2013; Tarlow and Zemel, 2012; Rosenfeld et al., 2014; Keshet, 2014). They have also been shown to be powerful when applied to latent variable models when optimizing for decomposable loss functions (Wang and Mori, 2011; Felzenszwalb et al., 2010; Yu and Joachims, 2009).

Our large-margin method learns latent variable models via optimizing non-decomposable loss functions. It interleaves the Concave-Convex Procedure (CCCP) (Yuille and Rangarajan, 2001) for populating latent variables with dual decomposition (Komodakis et al., 2011; Rush and Collins, 2012). The latter factorizes the hard optimization problem (encountered in learning) into smaller independent sub-problems over the training instances. We then present linear programming and local search methods for effective optimization of the sub-problems encountered in the dual decomposition. Our local search algorithm leads to a speed up of 7,000 times compared to the exhaustive search used in the literature (Joachims, 2005; Ranjbar et al., 2012).

Our work is the first to make use of max-margin training in distant supervision of relation extraction models. We demonstrate the effectiveness of our proposed method compared to two strong baseline systems which optimize for the error rate and conditional likelihood, including a state-of-the-art system by Hoffmann et al. (2011). On several data conditions, we show that our method outperforms the baseline and results in up to 8.5% improvement in the F_1-score.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 892–900,
Denver, Colorado, May 31 – June 5, 2015.

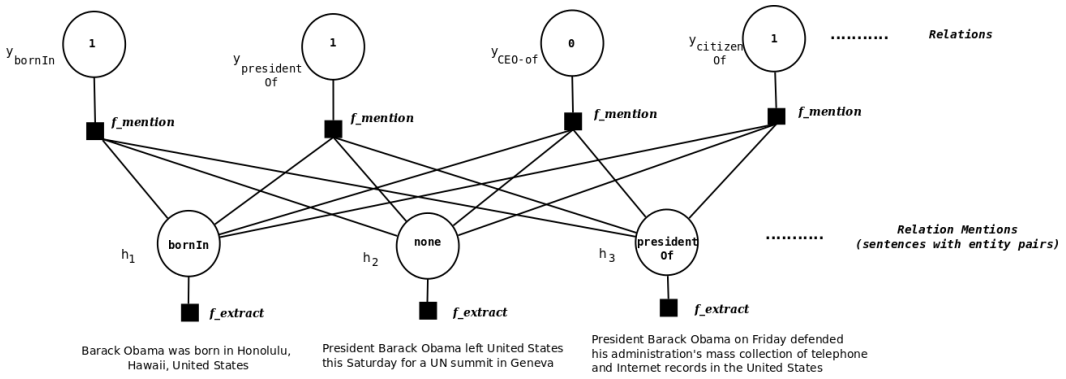

Figure 1: Graphical model instantiated for entity pair $\mathbf{x} := (\texttt{Barack Obama, United States})$

2 Preliminaries

2.1 Distant Supervision for Relation Extraction

Our framework is motivated by distant supervision for learning relation extraction models (Mintz et al., 2009). The goal is to learn relation extraction models by aligning facts in a database to sentences in a large unlabeled corpus. Since the individual sentences are not hand labeled, the facts in the database act as "weak" or "distant" labels, hence the learning scenario is termed as distantly supervised.

Prior work casts this problem as a multi-instance multi-label learning problem (Hoffmann et al., 2011; Surdeanu et al., 2012). It is multi-instance since for a given entity-pair, only the label of the bag of sentences containing both entities (aka mentions) is given. It is multi-label since a bag of mentions can have multiple labels. The inter-dependencies between relation labels and (hidden) mention labels are modeled by a Markov Random Field (Figure 1) (Hoffmann et al., 2011). The learning algorithms used in the literature for this problem optimize the (conditional) likelihood, but the evaluation measure is commonly the F-score.

Formally, the training data is $\mathcal{D} := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathcal{X}$ is the entity-pair, $\mathbf{y}_i \in \mathcal{Y}$ denotes the relation labels, and $\mathbf{h}_i \in \mathcal{H}$ denotes the *hidden* mention labels. The possible relation labels for the entity pair are observed from a given knowledge-base. If there are L candidate relation labels in the knowledge-base, then $\mathbf{y}_i \in \{0,1\}^L$, (e.g. $y_{i,\ell}$ is 1 if the relation ℓ is licensed by the knowledge-base for the entity-pair) and $\mathbf{h}_i \in \{1, .., L, nil\}^{|\mathbf{x}_i|}$ (i.e. each mention realizes one of the relation labels or *nil*).

Notation. In the rest of the paper, we denote the collection of all entity-pairs $\{\mathbf{x}_i\}_{i=1}^N$ by $\mathbf{X} \in \boldsymbol{\mathcal{X}} := \mathcal{X} \times .. \times \mathcal{X}$, the collection of mention relations $\{\mathbf{h}_i\}_{i=1}^N$ by $\mathbf{H} \in \boldsymbol{\mathcal{H}} := \mathcal{H} \times .. \times \mathcal{H}$, and the collection of relation labels $\{\mathbf{y}_i\}_{i=1}^N$ by $\mathbf{Y} \in \boldsymbol{\mathcal{Y}} := \mathcal{Y} \times .. \times \mathcal{Y}$.

The aim is to learn a parameter vector $\mathbf{w} \in \mathbb{R}^d$ by which the relation labels for a new entity-pair $\mathbf{x}$ can be predicted

$$f_{\mathbf{w}}(\mathbf{x}) := \arg\max_{\mathbf{y}} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{h}, \mathbf{y}) \qquad (1)$$

where $\Phi \in \mathbb{R}^d$ is a feature vector defined according to the Markov Random Field, modeling the inter-dependencies between $\mathbf{x}$ and $\mathbf{y}$ through $\mathbf{h}$ (Figure 1). In training, we would like to minimize the loss function Δ by which the model will be assessed at test time. For the relation extraction task, the loss can be considered to be the negative of the F_β score:

$$F_\beta = \frac{1}{\frac{\beta}{\text{Precision}} + \frac{1-\beta}{\text{Recall}}} \qquad (2)$$

where $\beta = 0.5$ results in optimizing against F_1-score. Our proposed learning method optimizes those loss functions Δ which cannot be decomposed over individual training instances. For example, F_β depends non-linearly on Precision and Recall which in turn require the predictions for *all* the entity pairs in the training set, hence it cannot be decomposed over individual training instances.

2.2 Structured Prediction Learning

The goal of our learning problem is to find $\mathbf{w} \in R^d$ which minimizes the expected loss, aka *risk*, over a

new sample $\mathcal{D}'$ of size N':

$$R^{\Delta}_{f_{\mathbf{w}}} := \int \Delta\Big(\big(f_{\mathbf{w}}(\mathbf{x}'_1), .., f_{\mathbf{w}}(\mathbf{x}'_{N'})\big), \big(\mathbf{y}'_1, .., \mathbf{y}'_{N'}\big)\Big) dPr(\mathcal{D}') \tag{3}$$

Generally, the loss function Δ cannot be decomposed into a linear combination of a loss function δ over individual training samples. However, most discriminative large-margin learning algorithms assume for simplicity that the loss function is decomposable and the samples are i.i.d. (independent and identically distributed), which simplifies the sample risk $R^{\Delta}_{f_{\mathbf{w}}}$ as:

$$R^{\delta}_{f_{\mathbf{w}}} := \int \delta(f_{\mathbf{w}}(\mathbf{x}'), \mathbf{y}') dPr(\mathbf{x}', \mathbf{y}') \tag{4}$$

Often learning algorithms make use of the empirical risk as an approximation of the sample risk:

$$\hat{R}^{\delta}_{f_{\mathbf{w}}} := \frac{1}{N} \sum_{i=1}^{N} \delta(f_{\mathbf{w}}(\mathbf{x}_i), \mathbf{y}_i) \tag{5}$$

For non-decomposable loss functions, such as F_β, Δ cannot be expressed in terms of instance-specific loss function δ to construct the empirical risk of the kind in Eq. (5). Instead, we need to optimize the empirical risk constructed based on the sample loss:

$$\hat{R}^{\Delta}_{f_{\mathbf{w}}} := \Delta\Big(\big(f_{\mathbf{w}}(\mathbf{x}_1), .., f_{\mathbf{w}}(\mathbf{x}_N)\big), \big(\mathbf{y}_1, .., \mathbf{y}_N\big)\Big) \tag{6}$$

or equivalently

$$\hat{R}^{\Delta}_{f_{\mathbf{w}}} := \Delta(f_{\mathbf{w}}(\mathbf{X}), \mathbf{Y}) \tag{7}$$

where $f_{\mathbf{w}}(\mathbf{X}) := (f_{\mathbf{w}}(\mathbf{x}_1), .., f_{\mathbf{w}}(\mathbf{x}_N))$.

Having defined the empirical risk in Eq (7), we formulate the learning problem as a structured prediction problem. Instead of learning a mapping function $f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$ from an individual instance $\mathbf{x} \in \mathcal{X}$ to its label $\mathbf{y} \in \mathcal{Y}$, let us learn a mapping function $\mathbf{f} : \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}$ from all instances $\mathbf{X} \in \boldsymbol{\mathcal{X}}$ to their labels $\mathbf{Y} \in \boldsymbol{\mathcal{Y}}$. We then define the best labeling using a linear discriminant function:

$$\mathbf{f}(\mathbf{X}) := \arg\max_{\mathbf{Y}' \in \boldsymbol{\mathcal{Y}}} \max_{\mathbf{H} \in \boldsymbol{\mathcal{H}}} \Big\{ \mathbf{w} \cdot \Psi(\mathbf{X}, \mathbf{H}, \mathbf{Y}') \Big\} \tag{8}$$

where $\Psi(\mathbf{X}, \mathbf{H}, \mathbf{Y}') := \sum_{i=1}^{N} \Phi(\mathbf{x}_i, \mathbf{h}_i, \mathbf{y}'_i)$. Based on the margin re-scaling formulation of structured prediction problems (Tsochantaridis et al., 2004), the training objective can be written as the following unconstrained optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C \max_{\mathbf{Y}'} \Big\{ \max_{\mathbf{H}} \mathbf{w} \cdot \Psi(\mathbf{X}, \mathbf{H}, \mathbf{Y}') - \max_{\mathbf{H}} \mathbf{w} \cdot \Psi(\mathbf{X}, \mathbf{H}, \mathbf{Y}) + \Delta(\mathbf{Y}', \mathbf{Y}) \Big\} \tag{9}$$

which is similar to the training objective for the latent SVMs (Yu and Joachims, 2009), with the difference that instance-dependent loss function δ is replaced by the sample loss function Δ. Learning $\mathbf{w}$ by optimizing the above objective function is challenging, and is the subject of the next section.

3 Optimizing the Training Objective

In this section we present our method to learn latent SVMs with non-decomposable loss functions. Our training objective is Eq (9), which can be equivalently expressed as:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C \max_{\mathbf{y}'_1, .., \mathbf{y}'_N} \Big\{ \Delta\Big((\mathbf{y}_1, .., \mathbf{y}_N), (\mathbf{y}'_1, .., \mathbf{y}'_N)\Big) + \sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}'_i) - \sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}_i) \Big\} \tag{10}$$

The training objective is non-convex, since it is the difference of two convex functions. In this section we make use of the CCCP to populate the hidden variables (Yu and Joachims, 2009; Yuille and Rangarajan, 2001), and interleave it with dual decomposition (DD) to solve the resulting intermediate loss-augmented inference problems (Ranjbar et al., 2012; Rush and Collins, 2012; Komodakis et al., 2011).

3.1 Concave-Convex Procedure (CCCP)

The CCCP (Yuille and Rangarajan, 2001) gives a general iterative method to optimize those non-convex objective functions which can be written as the difference of two convex functions $g_1(\mathbf{w}) - g_2(\mathbf{w})$. The idea is to iteratively lowerbound g_2 with a linear function $g_2(\mathbf{w}^{(t)}) + \mathbf{v} \cdot (\mathbf{w} - \mathbf{w}^{(t)})$, and take the following step to update $\mathbf{w}$:

$$\mathbf{w}^{(t+1)} := \arg\min_{\mathbf{w}} \big\{ g_1(\mathbf{w}) - \mathbf{w} \cdot \mathbf{v}^{(t)} \big\} \tag{11}$$

In our case, the training objective Eq (10) is the difference of two convex functions, where the second function g_2 is $C \sum_{i=1}^{N} \max_{\mathbf{h}} \big\{ \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}_i) \big\}$. The

Algorithm 1 The Training Algorithm (Optimizing Eq 10)

procedure OPT-LATENTSVM($\mathbf{X}, \mathbf{Y}$)
  Initialize $\mathbf{w}^{(0)}$ and set $t = 0$
  repeat
    for $i := 1$ to N **do**
      $\mathbf{h}_i^* := \arg\max_{\mathbf{h}} \mathbf{w}^{(t)} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}_i)$
    // Optimizing Eq 12
    $\mathbf{w}^{(t+1)} :=$ optSVM($\mathbf{X}, \mathbf{H}^*, \mathbf{Y}$)
    $t := t + 1$
  until some stopping condition is met
  return $\mathbf{w}^{(t)}$

lowerbound of $g_2(\mathbf{w})$ involves populating the hidden variables by:

$$\mathbf{h}_i^* := \arg\max_{\mathbf{h}} \left\{ \mathbf{w}^{(t)} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}_i) \right\}.$$

Therefore, in each iteration of our CCCP-based algorithm we need to optimize Eq (12), which is reminiscent of the standard structural SVM without latent variables:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C \max_{\mathbf{y'}_1,..,\mathbf{y'}_N} \left\{ \Delta\Big((\mathbf{y}_1,..,\mathbf{y}_N), (\mathbf{y'}_1,..,\mathbf{y'}_N) \Big) + \sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}'_i) - \sum_{i=1}^{N} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}_i^*, \mathbf{y}_i) \right\} \quad (12)$$

The above objective function can be optimized using the standard cutting-plane algorithms for structural SVM (Tsochantaridis et al., 2004; Joachims, 2005). The cutting-plane algorithm in turn needs to solve the *loss-augmented inference*, which is the subject of the next sub-section. The CCCP-based training algorithm is summarized in Algorithm 1.

3.2 Loss-Augmented Inference

To be able to optimize the training objective Eq (12) encountered in each iteration of Algorithm 1, we need to solve (the so-called) loss-augmented inference:

$$\max_{\mathbf{y}'_1,..,\mathbf{y}'_N} \quad \Delta\Big((\mathbf{y}_1,..,\mathbf{y}_N), (\mathbf{y}'_1,..,\mathbf{y}'_N) \Big) + \sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}'_i) \quad (13)$$

We make use of the dual decomposition (DD) technique to decouple the two terms of the above objective function, and efficiently find an approximate solution. DD is shown to be an effective technique for loss-augmented inference in structured prediction models *without* hidden variables (Ranjbar et al., 2012).

To apply DD to the loss-augmented inference (13), let us rewrite it as a constrained optimization problem:

$$\max_{\mathbf{y}'_1,..,\mathbf{y}'_N,\mathbf{y}''_1,..,\mathbf{y}''_N} \Delta\Big((\mathbf{y}_1,\ldots,\mathbf{y}_N), (\mathbf{y}'_1,\ldots,\mathbf{y}'_N) \Big) + \sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}_i{}'')$$

subject to

$$\forall i \in \{1,\ldots,N\}, \forall \ell \in \{1,\ldots,L\}, \quad y'_{i,\ell} = y''_{i,\ell}$$

Introduction of the new variables $(\mathbf{y}''_1, .., \mathbf{y}''_N)$ decouples the two terms in the objective function, and as we will see, leads to an effective optimization algorithm. After forming the Lagrangian, the dual objective function is derived as:

$$L(\mathbf{\Lambda}) := \max_{\mathbf{Y}'} \Delta(\mathbf{Y}, \mathbf{Y}') + \sum_i \sum_\ell \lambda_i(\ell) y'_{i,\ell} + \max_{\mathbf{Y}''} \sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}_i{}'') - \sum_i \sum_\ell \lambda_i(\ell) y''_{i,\ell}$$

where $\mathbf{\Lambda} := (\boldsymbol{\lambda}_1, .., \boldsymbol{\lambda}_N)$, and $\boldsymbol{\lambda}_i$ is a vector of Lagrange multipliers for L binary variables each of which represent a relation label. The two optimization problems involved in the dual $L(\mathbf{\Lambda})$ are independent and can be solved separately. The dual is an upperbound on the loss-augmented objective function for any value of $\mathbf{\Lambda}$; therefore, we can find the tightest upperbound as an approximate solution:

$$\min_{\mathbf{\Lambda}} L(\mathbf{\Lambda})$$

The dual is non-differentiable at those points $\mathbf{\Lambda}$ where either of the two optimisation problems has multiple optima. Therefore, it is optimized using the subgradient descent method:

$$\Lambda^{(t)} := \Lambda^{(t-1)} - \eta^{(t)} (\mathbf{Y}'_* - \mathbf{Y}''_*)$$

where $\eta^{(t)} = \frac{1}{\sqrt{t}}$ is the step size[1], and

$$\mathbf{Y}'_* := \arg\max_{\mathbf{Y}'} \Delta(\mathbf{Y}, \mathbf{Y}') + \sum_i \sum_\ell \lambda_i^{(t-1)}(\ell) y'_{i,\ell} \quad (14)$$

$$\mathbf{Y}''_* := \arg\max_{\mathbf{Y}''} \sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{h}, \mathbf{y}_i{}'') - \sum_i \sum_\ell \lambda_i^{(t-1)}(\ell) y''_{i,\ell} \quad (15)$$

[1]Other (non-increasing) functions of the iteration number t are also plausible, as far as they satisfy the following conditions (Komodakis et al., 2011) needed to guarantee the convergence to the optimal solution in the subgradient descent method: $\eta^{(t)} \geq 0, \lim_{t\to\infty} \eta^{(t)} = 0, \sum_{t=1}^{\infty} \eta^{(t)} = \infty$

Algorithm 2 Loss-Augmented Inference

procedure OPT-LOSSAUG($\mathbf{w}, \mathbf{X}, \mathbf{Y}$)
  Initialize $\mathbf{\Lambda}^{(0)}$ and set $t = 0$
  repeat
    $\mathbf{Y}^{'}_{*} :=$ opt-LossLag($\mathbf{\Lambda}, \mathbf{Y}$) // Eq (14)
    $\mathbf{Y}^{''}_{*} :=$ opt-ModelLag($\mathbf{\Lambda}, \mathbf{X}$) // Eq (15)
    if $\mathbf{Y}^{'}_{*} = \mathbf{Y}^{''}_{*}$ **then**
      return $\mathbf{Y}^{'}_{*}$
    for $i := 1$ to N **do**
      for $\ell := 1$ to L **do**
        $\lambda_i^{(t+1)}(\ell) := \lambda_i^{(t)}(\ell) - \eta^{(t)}(y'_{i,\ell} - y''_{i,\ell})$
  until some stopping condition is met
  return $\mathbf{Y}^{'}_{*}$

The DD algorithm to compute the loss-augmented inference is outlined in Algorithm 2. Now the challenge is how to solve the above two optimization problems, which is the subject of the following section.

3.3 Effective Optimization of the Dual

The two optimization problems involved in the dual are hard in general. More specifically, the optimization of the affine-augmented model score (in Eq. 15) is as difficult as the MAP inference in the underlying graphical model, which can be challenging for loopy graphs. For the graphical model underlying distant supervision of relation extraction (Fig 1), we formulate the inference as an ILP (integer linear program). Furthermore, we relax the ILP to LP to speed up the inference, in the expense of trading exact solutions with approximate solutions[2].

Likewise, the optimization of the affine-augmented multivariate loss (in Eq. 14) is difficult. This is because we have to search over the entire space of $\mathbf{Y}' \in \mathcal{Y}$, which is exponentially large $\mathcal{O}(2^{N*L})$. However, if the loss term Δ can be expressed in terms of some aggregate statistics over $\mathbf{Y}'$, such as false positives (FPs) and false negatives (FNs), the optimization can be performed efficiently. This is due to the fact that the number of FPs can range from zero to the size of negative labels, and the number of FNs can range from zero to the number of positive labels. Therefore, the loss term can take $\mathcal{O}(N^2L^2)$ different values which can be represented on a two-dimensional grid. Fixing FPs and FNs to a grid point, $\mathbf{\Lambda} \cdot \mathbf{Y}'$ is maximized with respect to $\mathbf{Y}'$. The grid point which has the best value for $\Delta(\mathbf{Y}, \mathbf{Y}') + \mathbf{\Lambda} \cdot \mathbf{Y}'$ will then give the optimal solution for Eq (14).

Algorithm 3 Finding $\mathbf{Y}'_*$: Local Search

procedure OPT-LOSSLAG($\mathbf{\Lambda}, \mathbf{Y}$)
  $(idx_1^n \dots idx_{\#neg}^n) \leftarrow$ Sort $\downarrow$ $(\lambda_i(\ell))$ // FPs
  $(idx_1^n \dots idx_{\#pos}^n) \leftarrow$ Sort $\uparrow$ $(\lambda_i(\ell))$ // FNs
  Initialise (fp, fn) on the grid
  repeat
    for $((fp', fn') \in$ `Neigbours`(fp, fn) **do**
      $loss_{(fp',fn')} = \Delta(fp', fn') + \mathbf{\Lambda}_{sorted}$[3]
    $loss_{(fp'',fn'')} = \arg\max_{(fp',fn')} loss_{(fp',fn')}$
    if $loss_{(fp,fn)} > loss_{(fp'',fn'')}$ **then**
      break
    else
      $(fp, fn) = (fp'', fn'')$
  until $loss_{(fp,fn)} \leq loss_{(fp'',fn'')}$
return $\{ \mathbf{Y}'$ corresponding to $(fp, fn) \}$

Exhaustive search in the space of all possible grid points is not efficient as soon as the grid becomes large. Therefore, we have to adapt the techniques proposed in previous work (Ranjbar et al., 2012; Joachims, 2005). We propose a simple but effective *local search* strategy for this purpose. The procedure is outlined in Algorithm 3. We start with a random grid point, and move to the best neighbour. We keep hill climbing until there is no neighbour better than the current point. We define the neighbourhood by a set of exponentially-spaced points in all directions around the current point, to improve the exploration of the search space. We present some analysis on the benefits of using this search strategy vis-a-vis the exhaustive search in the Experiments section.

4 Experiments

Dataset: We use the challenging benchmark dataset created by Riedel et al. (2010) for distant supervision of relation extraction models. It is created by aligning relations from *Freebase*[4] with the sentences in *New York Times* corpus (Sandhaus, 2008). The labels for the datapoints come from the Freebase

[2]We observed in our experiments that relaxing the ILP to LP does not hurt the performance, but significantly speeds up the inference.

[3]For a given (fp, fn), we set $\mathbf{y}'$ by picking the sorted unary terms that maximize the score according to $\mathbf{y}$.

[4]`www.freebase.com`

database but Freebase is incomplete (Ritter et al., 2013). So a data point is labeled *nil* when either no relation exists or the relation is absent in Freebase. To avoid this ambiguity we train and evaluate the baseline and our algorithms on a subset of this dataset which consists of only non-nil relation labeled datapoints (termed as *positive dataset*). For the sake of completeness, we do report the accuracies of the various approaches on the entire evaluation dataset.

Systems and Baseline: Hoffmann et al. (2011) describe a state-of-the-art approach for this task. They use a perceptron-style parameter update scheme adapted to handle latent variables; their training objective is the conditional likelihood. Out of the two implementations of this algorithm, we use the better[5] of these two[6], as our baseline (denoted by `Hoffmann`). For a fair comparison, the training dataset and the set of features defined over it are common to all the experiments.

We discuss the results of two of our approaches. One, is the LatentSVM max-margin formulation with the simple decomposable Hamming loss function which minimizes the error rate (denoted by `MM-hamming`). The other is the LatentSVM max-margin formulation with the non-decomposable loss function which minimizes the negative of F_β score (denoted by `MM-F-loss`)[7].

Evaluation Measure: The performance measure is F_β which can be expressed in terms of false positives (FP) and false negatives (FN) as:

$$F_\beta = \frac{N_p - FN}{\beta(FP - FN) + N_p}$$

where β is the weight assigned to precision (and $1-\beta$ to recall). FP, FN and N_p are defined as :

$$\begin{aligned} FP &= \sum_i \sum_\ell y'_{i,\ell}(1 - y_{i,l}) \\ FN &= \sum_i \sum_\ell y_{i,\ell}(1 - y'_{i,l}) \\ N_p &= \sum_i \sum_\ell y_{i,\ell} \end{aligned}$$

[5]It is not quite clear why the performance of the two implementations are different.

[6]`nlp.stanford.edu/software/mimlre.shtml`

[7]We use a combination of F1 loss and hamming loss, as using only F1-loss overfits the training dataset, as observed from the experiments.

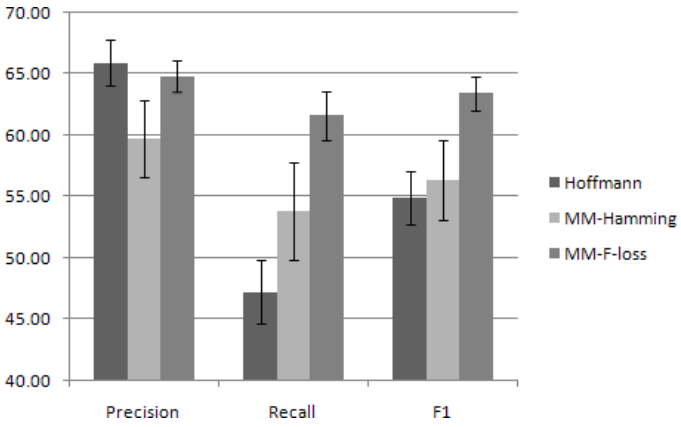

Figure 2: Experiments on 10% Riedel datasets.

	Precision	*Recall*	F_1
Hoffmann	65.93	47.22	54.91
MM-Hamming	59.74	53.81	56.32
MM-F-loss	64.81	61.63	63.44

Table 1: Average results on 10% Riedel datasets.

We use $1-F_\beta$ as the expression for the multivariate-loss.

4.1 Training on Sub-samples of Data

We performed a number of experiments using different randomized subsets of the Riedel dataset (10% of the positive dataset) for training the max-margin approaches. This was done in order to empirically determine a good set of parameters for training. We also compare the results of the approaches with `Hoffmann` trained on the same sub-samples.

Comparison with the Baseline: We report the average over 15 subsets of the dataset with a 90% confidence interval (using student-t distribution). The results of these experiments are shown in Figure 2 and Table 1. We observe that both `MM-hamming` and `MM-F-loss` have higher F_1-score compared to the baseline. There is a significant improvement in F_1-score to the tune of 8.52% for the multivariate performance measure over `Hoffmann`. There is also an improvement of F_1-score of 7.12% compared to `MM-Hamming`. This highlights the importance of using non-linear loss functions compared to simple loss functions like error rate during training.

However, `Hoffmann` has a marginally higher precision of about 1.13%. We noticed that this was

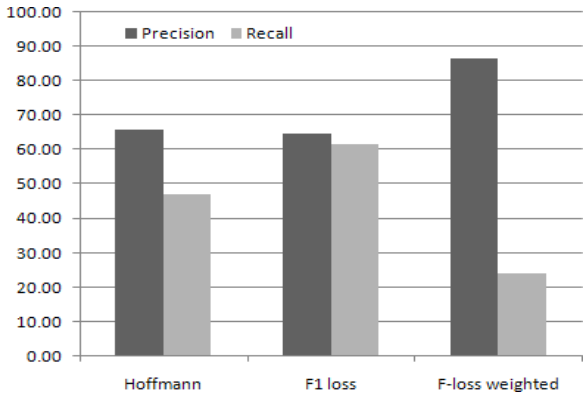

Figure 3: Weighting of Precision and Recall ($\beta = 0.833$)

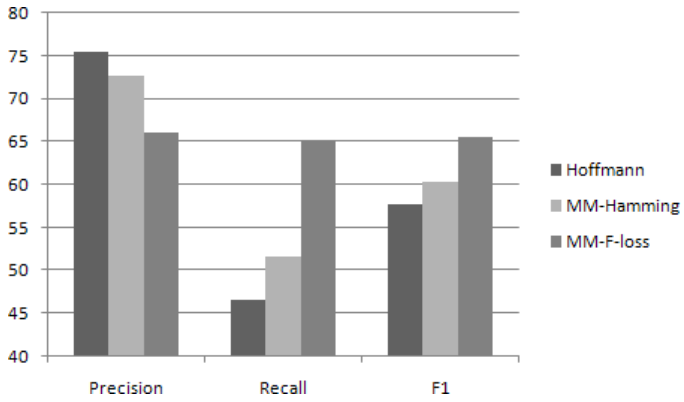

Figure 4: Overall accuracies Riedel dataset

due to over-fitting the data, as the performance on the training datasets were very high. One more interesting observation of `MM-F-loss` is that it is fairly balanced w.r.t both precision and recall which the other approaches do not exhibit.

Tuning towards Precision/Recall: Often we come across situations where either precision or recall is important for a given application. This is modeled by the notion of F_β (van Rijsbergen, 1979). One of the main advantages of using a non-decomposable loss function like F_β is the ability to vary the learning algorithm to factor such situations. For instance we can tune the objective to favor precision more than recall by "up-weighting" precision in the F_β-score.

For instance, in the previous case we observed that `MM-F-loss` has a marginally poorer precision compared to `Hoffmann`. Suppose we increase the weight of precision, $\beta = 0.833$, we observe a dramatic increase in precision from 65.83% to 86.59%. As expected, due to the precision-recall trade-off, we observe a decrease in recall. The results are shown in Figure 3.

Local vs. Exhaustive Grid Search: As we described in Section 3.3, we devise a simple yet efficient local search strategy to search the space of (FP, FN) grid-points. This enables a speed up of three orders of magnitude in solving the dual-optimization problem. In Table 2, we compare the average time per iteration and the F_1-score when each of these techniques is used for training on a sub-sample dataset. We observe that there is a significant decrease in training time when we use local search (almost 7000 times faster), with a negligible decrease in F_1-score (0.073%).

	avg. time per iter.	F_1
Local Search	0.09s	58.322
Exhaustive Search	630s	58.395

Table 2: Local vs. Exhaustive Search.

4.2 The Overall Results

Figure 4 and Table 3 present the overall results of our approaches compared to the baseline on the *positive* dataset. We observe that `MM-F-loss` has an increase in F_1-score to the tune of $\sim$8% compared to the baseline. This confirms our observation on the sub-sample datasets we saw earlier.

By assigning more weight to precision, we are able to improve over the precision of `Hoffmann` by $\sim$1.6% (Table 4). When precision is tuned with a higher weight during training of `MM-F-loss`, we see an improvement in precision without much dip in recall.

	Precision	*Recall*	F_1
Hoffmann	75.436	46.615	57.623
MM-Hamming	76.839	50.462	60.918
MM-F-loss	65.991	65.211	65.598

Table 3: Overal results on the *positive* dataset.

	Precision	*Recall*	F_β
Hoffmann	75.44	46.62	57.62
MM-F-loss-wt	77.04	53.44	63.11

Table 4: Increasing weight on Precision in F_β.

4.3 Discussion

So far we have discussed the performance of various approaches on the *positive* evaluation dataset. Our approach is shown to improve overall F_β-score having better recall than the baseline. By suitably tweaking the F_β we show an improvement in precision as well.

The performance of the approaches when evaluated on the entire test dataset (consisting of both nil and non-nil datapoints) is shown in Table 5. Max-margin based approaches generally perform well when trained only on the *positive* dataset compared to `Hoffmann`. However, our F_1-scores are $\sim$8% less when we train on the entire dataset consisting of both nil and non-nil datapoints.

Trained On$\rightarrow$	*entire dataset*	*positive dataset*
Hoffmann	23.14	3.269
MM-Hamming	13.20	16.26
MM-F-loss	13.94	21.93

Table 5: F_1-scores on the entire test set.

In a recent work, Xu et al. (2013) provide some statistics about the incompleteness of the Riedel dataset. Out of the sampled 1854 sentences from NYTimes corpus most of the entity pairs expressing a relation in Freebase correspond to false negatives. This is one of the reasons why we do not consider nil labeled datapoints during training and evaluation.

MIMLRE (Surdeanu et al., 2012) is another state-of-the-art system which is based on the EM algorithm. Since that system uses an additional set of features for the relation variables $\mathbf{y}$, it is not our primary baseline. On the *positive* dataset, our model `MM-F-loss` achieves a F_1-score of 65.598% compared to 65.341% of MIMLRE. As part of the future work, we would like to incorporate the additional features present in MIMLRE into our approach.

5 Conclusion

In this paper, we described a novel max-margin approach to optimize non-linear performance measures, such as F_β, in distant supervision of information extraction models. Our approach is general and can be applied to other latent variable models in NLP. Our approach involves solving the hard-optimization problem in learning by interleaving Concave-Convex Procedure with dual decomposition. Dual decomposition allowed us to solve the hard sub-problems independently. A key aspect of our approach involves a local-search algorithm which has led to a speed up of 7,000 times in our experiments. We have demonstrated the efficacy of our approach in distant supervision of relation extraction. Under several conditions, we have shown our technique outperforms very strong baselines, and results in up to 8.5% improvement in F_1-score.

For future work, we would like to maximize other performance measures, such as area under the curve, for information extraction models. Furthermore, we would like to explore our approach for other latent variable models in NLP, such as those in machine translation.

Acknowledgements

Gholamreza Haffari is grateful to National ICT Australia (NICTA) for their generous funding, as part of the Machine Learning Collaborative Research Projects. Ajay Nagesh acknowledges Xerox Research Centre India (XRCI) for their travel support in the form of International Student Travel grant.

References

Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 541–550, Stroudsburg, PA, USA. Association for Computational Linguistics.

T. Joachims. 2005. A support vector method for multivariate performance measures. In *International Conference on Machine Learning (ICML)*, pages 377–384.

Joseph Keshet. 2014. Optimizing the measure of performance in structured prediction. In Jeremy Jancsary Sebastian Nowozin, Peter V. Gehler and Christoph H. Lampert, editors, *Advanced Structured Prediction*. The MIT Press.

Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. 2011. Mrf energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):531–552.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mani Ranjbar, Arash Vahdat, and Greg Mori. 2012. Complex loss optimization via dual decomposition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2304–2311.

Mani Ranjbar, Tian Lan, Yang Wang, Stephen N. Robinovitch, Ze-Nian Li, and Greg Mori. 2013. Optimizing nondecomposable loss functions in structured prediction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(4):911–924.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III*, ECML PKDD'10, pages 148–163, Berlin, Heidelberg. Springer-Verlag.

Alan Ritter, Luke Zettlemoyer, Mausam, and Oren Etzioni. 2013. Modeling missing data in distant supervision for information extraction. *TACL*, 1:367–378.

Nir Rosenfeld, Ofer Meshi, Amir Globerson, and Daniel Tarlow. 2014. Learning structured models with the auc loss and its generalizations. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Alexander M. Rush and Michael Collins. 2012. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. *J. Artif. Intell. Res. (JAIR)*, 45:305–362.

E. Sandhaus. 2008. The new york times annotated corpus.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 455–465, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel Tarlow and Richard S Zemel. 2012. Structured output learning with high order loss functions. In *Proceedings of the 15th Conference on Artificial Intelligence and Statistics*.

Benjamin Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *NIPS*.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning (ICML)*, pages 104–112.

C. J. van Rijsbergen. 1979. *Information retrieval*. Butterworths, London, 2 edition.

Yang Wang and Greg Mori. 2011. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(7):1310–1323.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 665–670, Sofia, Bulgaria, August. Association for Computational Linguistics.

Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, page 147.

Alan L. Yuille and Anand Rangarajan. 2001. The concave-convex procedure (cccp). In *NIPS*, pages 1033–1040.